



**FACULTAD DE INFORMATICA**  
**UNIVERSIDAD COMPLUTENSE DE MADRID**

---

**SISTEMA INTELIGENTE DE DETERMINACIÓN  
DE PLANTAS DE LA FLORA IBERICA**

**ASIGNATURA: SISTEMAS INFORMATICOS**  
**2007- 2008**

---

**MIEMBROS DEL PROYECTO:**

**LORENA PONCE MARTINEZ**  
**ELENA RODRÍGUEZ VALLE**  
**MARIA CEDENA MOYANO**

**DIRECTOR DEL PROYECTO:**

**LUIS GARMENDIA SALVADOR**

**CODIRECTOR DEL PROYECTO:**

**ALFONSO GARMENDIA SALVADOR**

**DEPARTAMENTO:**

**DEPARTAMENTO ISIA**

## **INDICE**

<b>1. INTRODUCCIÓN.....</b>	<b>4</b>
<b>2. ESTADO DEL ARTE.....</b>	<b>6</b>
<b>a. FLORA.....</b>	<b>6</b>
<b>b. FLOR.....</b>	<b>8</b>
<b>c. FAMILIA.....</b>	<b>9</b>
<b>d. GÉNERO.....</b>	<b>10</b>
<b>e. PLANTA.....</b>	<b>10</b>
<b>f. ESPECIE.....</b>	<b>10</b>
<b>g. VEGETACIÓN.....</b>	<b>10</b>
<b>h. VEGETAL.....</b>	<b>11</b>
<b>3. DETERMINACIONES.....</b>	<b>12</b>
<b>a. PRIMERA DETERMINACIÓN.....</b>	<b>14</b>
<b>b. SEGUNDA DETERMINACIÓN.....</b>	<b>28</b>
<b>c. TERCERA DETERMINACIÓN.....</b>	<b>34</b>
<b>4. HERRAMIENTA NITIDA.....</b>	<b>38</b>
<b>a. HERRAMIENTA INICIAL.....</b>	<b>40</b>
<b>b. CARACTERÍSTICAS SELECCIONADAS.....</b>	<b>43</b>
<b>c. EXPLICACIÓN DE LAS CARACTERÍSTICAS.....</b>	<b>44</b>
<b>d. CARACTERÍSTICAS AÑADIDAS POR EL         EXPERTO.....</b>	<b>77</b>
<b>e. EXPLICACIÓN DE LA HERRAMIENTA         NÍTIDA.....</b>	<b>90</b>

f.	EJECUCIÓN DE LA HERRAMIENTA.....	106
g.	NOVEDAD DE LA HERRAMIENTA.....	112
5.	HERRAMIENTA BORROSA.....	113
a.	LOGICA BORROSA.....	114
b.	HERRAMIENTA XFUZZY.....	130
c.	EXPLICACIÓN DE LA HERRAMIENTA BORROSA.....	154
d.	NOVEDAD EN LA HERRAMIENTA.....	166
e.	EJECUCIÓN DE LA HERRAMIENTA.....	171
6.	HERRAMIENTA BASADA EN EL CONOCIMINETO DEL EXPERTO.....	178
a.	EXPLICACIÓN DE LA HERRAMIENTA.....	180
b.	EJECUCIÓN DE LA HERRAMIENTA.....	190
7.	APÉNDICE: HERRAMIENTA XFUZZY.....	198
8.	APÉNDICE: PROBLEMA CON XFUZZY.....	222
9.	APÉNDICE:COMPARAR REGLAS XFUZZY CON LAS MODIFICADAS.....	224
10.	PALABRAS CLAVE.....	227
11.	BIBLIOGRAFÍA.....	228

# INTRODUCCION

El objetivo de este proyecto es la determinación de las plantas pertenecientes a la flora ibérica, proporcionando una ayuda adicional a los biólogos para dicho fin.

Se trata de una herramienta visual en la cual introduciendo una serie de características propias de la planta a determinar conseguimos mostrar al usuario la familia a la que pertenece, o una serie de familias a las que puede pertenecer centrando el estudio según los grados de pertenencia gracias a las reglas de inferencia de los sistemas borrosos.

Este proyecto lo hemos realizado en colaboración con la Universidad Politécnica de Valencia, especialmente con un experto biólogo llamado Alfonso Garmendia Salvador y su grupo de alumnos.

## ABSTRACT

The goal of this project is the classification of the plants belonging to the Iberian's flora, providing an additional help to the biologists for the mentioned purpose.

It is a visual tool in which, introducing a series of characteristics of the plant you want to classify, it manages to show to the user the family which the plant belongs to, or a series of families which it can belong to, centring the study according to the degrees of belonging due to the rules of inference of the blurry systems.

This project has been realized in collaboration with the Technical University of Valencia, specially with an expert so called biologist Alfonso Garmendia Salvador and his group of pupils.

Autorizamos a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria como el código, la documentación y/o el prototipo desarrollado.

Firmado:

Lorena Ponce Martínez

Elena Rodríguez Valle

Maria Cedená Moyano

## ESTADO DEL ARTE

Nuestro proyecto tiene como objeto de estudio la flora ibérica, por lo tanto como nuestro conocimiento en esa materia es relativamente nulo debemos de comenzar realizando un estudio de todo lo referente a la flora ibérica.

Previamente a este proyecto nosotras no teníamos conocimiento técnico de este campo por lo que teníamos que primeramente realizar un estudio de la terminología técnica de dicho campo.

Lo primero que vamos a hacer es una descripción general del campo en el que se centra nuestro proyecto.

Inicialmente comenzaremos con una explicación acerca del significado de flora.

### FLORA

La flora es un término procedente del latín que alude a la diosa romana Flora, diosa de las flores, jardines y primavera.

La flora en botánica se considera como el conjunto de plantas que pueblan un país. También se puede referir a otra extensión distinta como una península, un continente, una sierra, distintos climas,...

También se proporciona una descripción de las plantas que existen así como la época de floración que posee cada una y la cantidad de dicha planta que existe en la extensión sobre la que nos refiramos.

El conjunto de plantas que se pueden encontrar en las diferentes extensiones a las que nos hemos referido anteriormente se deben a un periodo geológico en concreto o que son propias de un ecosistema determinado.

La flora a diferencia de la vegetación atiende al número de especies, al contrario que la vegetación que atiende a la distribución de estas por la extensión a la que nos estamos refiriendo y a la importancia de esta con respecto a la cantidad de especies residente en la extensión y el tamaño de estas.

Como conclusión obtenemos que la vegetación esta determinada por el clima y demás factores ambientales que se producen en la flora.

Hay otras disciplinas botánicas como pueden ser la geobotánica o la fitogeografía que realizan el estudio de la distribución de los vegetales sobre la Tierra. Realizando dicho estudio mediante el análisis de la distribución y obteniendo sus características. También analizan las causas que los condicionan y las leyes a las que responde.

Esta disciplina analiza la unión entre el mundo vegetal y el mundo terrestre. Como cita mencionar:

"La ciencia que estudia el hábitat de las plantas en la superficie terrestre" Huguet del Villar.

La fitogeografía se conoce como el estudio de las formaciones vegetales más conocidas como florística.

### **Otras descripciones:**

La flora es un conjunto de especies vegetales de un territorio determinado. Este puede tener mayor o menor extensión, y en este sentido se habla de flora local, comarcal, regional, etc...

En las disciplinas de botánica y paleontología las hipótesis paleontológicas referentes a la distribución de los continentes en épocas anteriores se apoyan de forma ostensible en el reparto de las flores actuales y pasadas sobre la Tierra. Así, la hipótesis de Gondwana, supuesto continente austral de la era primaria, separado del continente nordatlántico por un gran mar llamado Tetis, se apoyo en el descubrimiento de *Glossopteris* (primitiva gimnosperma fósil) en las tierras que habría ocupado el citado continente. Sin embargo, el hallazgo de fósiles de esta planta en el Norte de Rusia opuso serias dudas en la hipótesis expuesta. A pesar de ello, y dada la facilidad de diseminación de los vegetales, se puede afirmar de manera general que las flores fósiles aportan buenos datos paleoclimatológicos, pero son de dudosa utilidad en paleogeografía. Así, la línea Wallace, no existe para un botánico. En efecto, el estrecho de Macasar, por el que discurre, no es obstáculo suficiente para impedir la dispersión de las plantas, seres especialmente aptos para la diseminación a grandes distancias.

La hipótesis sobre la benignidad del clima en el Ártico primario esta refrendada por el hallazgo de una rica flora permocarbonífera de tipo cálido en varios territorios

circumboreales: Spitzberg, Tierra de Ellesmere y Alaska. La exuberante flora carbonífera que ha dado lugar a muchos yacimientos de carbón estaba constituida por pteridofitos (equisetos, calamites, helechos arborescentes, ...) y gimnospermas primitivas. El reparto cosmopolita de los vegetales en el triásico, jurásico y cretáceo induce a creer en una uniformidad del clima alrededor del globo.

Durante el terciario se observa un cambio importante en las floras de la Tierra. Mientras que en el primario y el secundario aquellas son de tipo cálido o templado, al entrar en el eoceno aparecen especies de carácter menos termófilo. En Alaska y Siberia del norte, bosques de *Metasequoia*. En Groenlandia, *Taxodium*, *betula*, *ulmus*, .. En Europa central, *Ginkgo*, *sequoia*, *ulmus*, ... Un carácter general a lo largo del terciario es la progresión hacia el S. De las floras del ártico hasta más allá de los 45° de latitud. América del sur, que estuvo prácticamente aislada del Norte desde el paleoceno superior hasta el plioceno, posee una flora terciaria propia caracterizada por *Nothofagus* (hayas del sur) y *Araucaria*.

En Australia se encuentran en este periodo *Nothofagus* y *Eucalyptus*. Durante las glaciaciones cuaternarias las floras realizaron migraciones oscilantes. Ocupaban zonas determinadas durante ciertos periodos favorables y desaparecían de ellas cuando las condiciones eran adversas. Ello ha dado lugar a la conservación de ciertas plantas de distribución septentrional en zonas meridionales donde se mantiene microclimas parecidos a los que reinaron en el territorio cuando tuvo lugar la expansión de estas especies. Un ejemplo lo constituyen las flores alpinas de los sistemas orográficos del Sur de Europa. Actualmente la diversidad de climas que se dan en la Tierra condiciona la existencia de numerosas floras que se agrupan en los reinos florales holártico, paleotropical, neotropical, australiano, capense y antártico.

Ahora vamos al término de flor para conocer mejor cuál es el objeto de nuestro proyecto.

## FLOR

Las flores son típicas de los espermatofitos (gimnospermas y angiospermas) aunque en algunos pteridofitos (selaginelas) existen ya agrupaciones de esporofilos que responden a la estructura floral típica. Las piezas florales se insertan de modo helicoidal (flor aciclica) o formando verticilos (flor cíclica) sobre un brote de crecimiento limitado que en las



angiospermas suele ser muy corto (tálamo). Las flores de las gimnospermas son casi siempre acíclicas y suelen carecer de envoltura. En las angiospermas las flores son generalmente cíclicas y a menudo presentan una envoltura estéril (perianto) que en el caso más complejo se compone de cáliz (formado por sépalos) y corola (integrada por pétalos). La porción fértil de la flor comprende los microsporofilos (estambres), que en conjunto reciben el nombre de androceo, y los macrosporofilos (carpelos), que constituyen el gineceo. Si existen ambos tipos de esporofilos, la flor es hermafrodita; si falta uno de ellos, se habla de flores unisexuales: masculinas si solo poseen androceo y femeninas si únicamente tiene gineceo. Según el número de verticilos existentes, se habla de flores tetracíclicas (4), pentacíclicas (5), etc. Según el número de piezas por verticilo, de flores dimeras (2), trímeras (3), tetrámeras (4), etc. Por lo que respecta al tipo de simetría que presentan, se distinguen flores radiadas o actinomorfas, bilaterales, dorsiventrales o zigomorfas y asimétricas. Una flor cíclica completa comprende, de fuera adentro, 4 o 5 verticilos: uno de sépalos (cáliz), uno de pétalos (corola), uno o dos de estambres (androceo) y uno de carpelos (gineceo). En algunos grupos de plantas y en ciertos casos teratológicos se multiplica o se reduce el número de piezas de determinados verticilos; la reducción de algunos elementos llevada a su extremo máximo conduce a la desaparición del verticilo, como ocurre en las flores desnudas (carentes de perianto) y estériles (carentes de esporofitos). El cáliz consta de un número variable de antófilos, que por lo común presentan color verde; en caso de presentar otra coloración y morfología semejante a la del siguiente verticilo se denominan sépalos petaloides. La corola, de color variable, está constituida por pétalos, antófilos de estructura particular que no presentan estomas en su epidermis. El androceo está constituido por los estambres (microsporofilos). El gineceo comprende uno o más carpelos (macrosporofilos); cuando existen varios carpelos, estos pueden ser libres o bien estar soldados en uno o más pistilos, cada carpelo o cada pistilo se compone de ovario, estilo y estigma.

## FAMILIA

Familia significa la categoría sistemática que sitúa entre el género y el orden, y comprende un conjunto de géneros que presentan caracteres comunes importantes. En botánica terminan la mayoría en -aceas.

## GÉNERO

Genero en biología significa categoría taxonómica intermedia entre la familia y la especie. Aunque los géneros suelen comprender varias especies estrechamente afines, hay también géneros monoespecíficos, es decir, que incluyen una solo especie. Los géneros designan un sustantivo latín o latinizado que forma el primer termino de la designación binomial de las especies.

## PLANTA

Planta en términos vegetales es un ser orgánico que crece y vive sin mudar de lugar por impulso voluntario.

## ESPECIE

Especie es el conjunto de seres vivos, semejantes en morfología, capaces de reproducirse sexualmente entre si, pero no con individuos pertenecientes a otros grupos análogos.

## VEGETACIÓN

Al describir la riqueza vegetal de un país de términos de vegetación, se atiende más al aspecto fisonómico y de interrelaciones entre las distintas plantas que a la enumeración sistemática de las mismas. Teniendo en cuenta su estructura, se distinguen varios tipos de comunidades vegetales: los bosques, formados por una cobertura vegetal fundamentalmente arbórea; las maquias y otras comunidades de monte bajo en las que los arbustos tienen papel preponderante; las poblaciones de matas y arbustos, que reciben distinta calificación según sea la especie predominante, y los prados, en los que tienen máxima importancia las hierbas. La estructura y composición florística de las comunidades vegetales de un país obedecen a la interacción de muchos factores distintos. Unos dependen de la situación geográfica y las características físicas del país. Otros derivan de la composición biótica de la comunidad. Entre los primeros cabe destacar la latitud y, especialmente, el clima cuyos rasgos fundamentales vienen determinados por aquella, la situación de la comunidad en un lugar determinado de la orografía de la región (en este sentido influyen sobremanera la altitud, las diferencias entre solana y umbría en los valles, las riberas, etc..) , y el sustrato

geológico en cuanto determinante del tipo de suelo. Entre los segundos debe citarse en primer lugar la estructura de la vegetación que permite, por ejemplo el crecimiento de plantas adaptadas al bosque e impide la colonización del mismo por especies más heliofilas. Otros tipos de interacción derivan de las modificaciones que producen sobre el sustrato los productos de ciertos vegetales en cuanto determinantes de unas especiales condiciones químicas. Los cambios geológicos han contribuido a menudo a limitar el área geográfica de muchas especies que podrían colonizar áreas más extensas atidiendo a la climatología actual.

## VEGETAL

Vegetal, perteneciente o relativo a las plantas.

Si bien resulta fácil distinguir las especies animales de los vegetales en las formas mas evolucionadas, al considerar grupos taxonómicos inferiores las diferencias son menores e incluso dudosas. De hecho, es imposible establecer una definición vegetal que los incluya a todos y excluya a los animales; animales y vegetación poseen una misma estructura física, igual composición química y una estructura celular coincidente. En esquema, las plantas típicas (verdes) se separan de los animales porque:

- Poseen clorofila

- Sus células están revestidas de una membrana rígida.

- Son autótrofas y, por tanto, capaces de aprovechar de modo inmediato la energía solar, y frente a los animales, que son consumidores, representan el nivel productor

- Tienen crecimiento ilimitado y constituyen, por ello un sistema abierto

- Son inmóviles. Sin embargo, ninguno de estos caracteres es, por si solo, suficiente, ya que existen vegetales carentes de clorofila, muchos grupos poseen células reproductoras móviles, no son raros los vegetales saprofitos y parásitos, etc.

# DETERMINACIONES

Existen en total unas 350.000 especies vegetales, de las cuales la mitad son plantas con flores (fanerógamas); una tercera parte corresponde a las algas y los hongos y el resto a los demás grupos. Dentro de la multitud de formas de organización de los vegetales, que incluyen desde seres unicelulares a organismos con tejidos y órganos bien diferenciados, se distinguen tres tipos fundamentales: protofitos, talofitos y cormofitos. Atendiendo a los fenómenos de sexualidad se admiten desde antiguo, y en un simple análisis superficial, dos grandes grupos: criptógamas y fanerógamas. Por su parte y de un modo de crecimiento, las plantas pueden distribuirse en los tres tipos ya establecidos popularmente de árboles, arbustos y hierbas. Si en vez de seguir criterios morfológicos se atiende a las relaciones de las plantas con el medio, pueden establecerse diversas series de grupos ecológicos, con las llamadas formas biológicas.

La necesidad de una determinación taxonómica general, que facilite el estudio del reino de las plantas y sirva para la identificación de cualquiera de las numerosas formas vegetales, resulta evidente. El intento más antiguo a este respecto lo debemos a Teofrasto (-370 a -285). Este autor dividía los vegetales en cuatro grupos: árboles, arbustos, matas y hierbas, y dentro de ellos distinguía familias (por ejemplo reconoció ya las umbelíferas); estableció asimismo el concepto de género como grupo de familias afines. Después de Teofrasto, no se alcanzó ningún progreso sensible en taxonomía vegetal hasta el siglo XVI. Diversos autores como Cesalpino (1583), J. Ray (1682-93) y Tournefort (1694), ensayaron sistemas taxonómicos artificiales, es decir, basados en caracteres fáciles de observar. Pero el sistema artificial más famoso es el establecido por Linneo en 1735, que se basa en el número y disposición de los órganos sexuales. Linneo distinguía 23 clases de plantas con flores y una clase, Cryptogamia, en la que incluía todos los demás vegetales (algas, hongos, etc.) y aun algunas plantas superiores de flores difíciles de reconocer. Posteriormente, y sobre todo desde que se impuso la teoría de la evolución, se ha tendido a hallar un sistema natural que no solo facilite una visión sintética, sino que, además, ponga de manifiesto el origen y las interrelaciones de los diversos grupos de plantas. El sistema de Q. Braun, publicado en 1864, concuerda en sus líneas generales con las concepciones modernas.

Dicha determinación fue perfeccionada luego por A. Eichler, A. Engler y R. Von Wettstein. Hasta principios de este siglo la determinación comúnmente admitida comprendía cinco grandes divisiones: esquizofitos (bacterias y cianofíceas), talofitos (algas y hongos), briofitos (musgos y hepáticas), pteridofitos (helechos, licopodios y equisetos) y espermatofitos (plantas con flores). En los sistemas taxonómicos posteriores se ha reconocido progresivamente una mayor diversidad dentro de las plantas inferiores, de suerte que los grupos de criptógamas han ido en aumento, mientras se ha reafirmado la uniformidad de las plantas superiores. Las determinaciones modernas admiten por lo menos siete divisiones: bacteriofitos (bacterias), cianofitos (algas azules), ficofitos (restantes algas), micofitos (hongos), briofitos, pteridofitos y espermatofitos. Actualmente, y de acuerdo sobre todo con caracteres fisiológicos y bioquímicos, se desglosa el grupo de los ficofitos en seis divisiones distintas: clorofitos (algas verdes), euglenofitos, crisofitos (diatomeas y otras algas inferiores), pirrofitos, feofitos (algas pardas) y rodofitos (algas rojas), y se separan los mixomicofitos de los hongos típicos. En contraste con la creciente complejidad atribuida a los vegetales inferiores, algunos autores reúnen los pteridofitos y los espermatofitos en un nuevo grupo (los tranqueofitos o plantas vasculares).

El desarrollo filogenético de las plantas ofrece aun muchos problemas. Sin embargo, la paleobotánica ha aportado las bases para el conocimiento de la historia evolutiva del reino vegetal y, con la ayuda de la morfología comparada, ha permitido esbozar un árbol genealógico de los vegetales (ciertamente incompleto y muy mejorable). Las plantas más antiguas conocidas son las cianofíceas correspondientes al precámbrico. En el cámbrico existían ya algas muy desarrolladas, y en el siluriano aparecieron las primeras plantas terrestres. La vegetación no resultó importante, sin embargo, hasta el devoniano. En el carbonífero prosperaban multitud de pteridofitos y diversas clases de gimnospermas; en cambio, las angiospermas no han dejado restos evidentes hasta el cretáceo inferior.

Por su tipo de vida autótrofa, los vegetales representan, como ya se ha dicho, el nivel productor, y por consiguiente su actividad resulta básica para el desarrollo de la vida animal. Capaces de aprovechar la energía lumínica para la síntesis de sustancias orgánicas, son, a su vez, la principal fuente de energía que hace posible la vida heterótrofa. El hombre mismo está supeditado, naturalmente, a la actividad del mundo vegetal y, aparte de sus relaciones directas con él, muchas plantas o productos derivados de ellas le son

directamente útiles. Existen unas mil especies vegetales de uso importante y general para el hombre. Entre ellas ocupan un lugar preeminente las plantas alimenticias, de las que se obtienen materiales usados en bruto (frutas, hojas, semillas, etc.) o después de una elaboración previa (aceite, azúcar, etc.) aunque no son estrictamente alimentos, deben añadirse aquí las especias (canela, clavo, azafrán, etc.) los aromatizantes (tomillo, ajedrea, enebro) y las bebidas (café, té, vinos, licores). Tiene asimismo mucho interés las plantas medicinales o los productos de ellas obtenidos (opio, quina, antibióticos, etc.), a los que pueden sumarse los narcóticos y estimulantes (cola, marihuana, tabaco, etc.). No pueden olvidarse tampoco los productos forestales (madera, corcho, resinas, etc.) las fibras textiles (algodón, yute, etc.), el caucho y otros exudados, y, como material de gran utilidad, la celulosa con la interminable lista de sus derivados (plástico, etc.). Son también productos vegetales, los procedentes de plantas que vivieron en otras épocas geológicas, el carbón, el petróleo y el gas natural. Otro aspecto de las plantas interesante para el hombre es su valor estético. Aunque en un futuro lejano el hombre pueda disminuir su actual dependencia del mundo vegetal, parece que la fuente de energía deberá buscarla simple en las plantas verdes.

Ahora explicaremos las distintas formas de determinación que hemos encontrado.

## PRIMERA DETERMINACIÓN

Una de ellas la obtenemos de la página web:

<http://www.rjb.csic.es/floraiberica/>

En ella encontramos una pequeña introducción que nos sirvió para poder comprender mejor los terminos usados en ella y que copiaremos textualmente a continuación:

*Flora ibérica* se ordena esencialmente a facilitar la identificación de las plantas vasculares autóctonas o naturalizadas en la Península Ibérica e Islas Baleares. Por eso, una gran parte de la información que suministra se refiere a los caracteres que diferencian unas plantas de otras y hemos pretendido que la estructura de las descripciones ayude a una contraposición rápida, fácil. Es decir, la obra no se concibió como prolijamente descriptiva, sino como diagnóstica en primer término. Ciertamente que la compleja taxonomía de algunos géneros nos ha llevado en no pocas ocasiones a descripciones más amplias, con lo que perdemos, en alguna medida, la homogeneidad en este punto.

Por lo que se refiere a las angiospermas, para las entidades taxonómicas elevadas y hasta el rango de familia inclusive, nos venimos ajustando en jerarquías y conceptos, prácticamente, al sistema de STEBBINS, *Flowering plants - Evolution above the species level* (1974). Nos apartamos del sistema en cuestión al aceptar como diversas las familias *Viscaceae* y *Loranthaceae* en el vol. VIII. Ya en el vol. I, nos habíamos apartado también al incluir en las *Papaveraceae* —como subfamilia *Fumarioideae*— las *Fumariaceae* de la referida obra y en la clave, al considerar familias independientes las *Asclepiadaceae*, *Lilaeaceae*, *Posidoniaceae* y *Amaryllidaceae*.

La terminología descriptiva se ajusta, salvo excepciones infrecuentes, a la de FONT QUER, *Diccionario de Botánica* (1953). Se puso especial empeño en evitar —aunque no siempre ha sido posible— los términos castellanos que vayan tal vez a resultarles incómodos a los usuarios portugueses de la obra.

Las descripciones de los géneros van a referirse de modo preferente a las plantas del territorio que nos ocupa. En tanto que las claves —así de familias como de géneros y especies o subespecies—, igualmente, se han basado en la observación directa y casi exclusiva de plantas ibéricas y baleáricas.

Para los nombres genéricos, tras las correspondientes autorías, nos limitamos a indicar, en su caso, "nom. cons.", lo que parecía útil y resultaba sencillo. Hemos decidido también, por añadidura no tan sencilla ni habitual, incluir entre corchetes, a continuación, la etimología de tales nombres, más indicaciones acerca de su género gramatical, prosodia y otras, cuando es posible hacer alguna útil en pocas palabras. Para los restrictivos específicos y subespecíficos, la oportuna información se da en el **Apéndice IV** de este sitio web, adelantándose tan solo en el cuerpo de la obra la muy parcial de que se habla líneas más abajo.

### **Información que se ofrece**

Sobre cada especie o subespecie damos las informaciones que siguen, por su orden riguroso:

1.º **Nombre considerado correcto** (es decir, el que según el Código Internacional de Nomenclatura Botánica debería utilizarse, a nuestro juicio taxonómico), **autor o autores** implicados en tal combinación, más **referencia bibliográfica** del texto que la establece con validez jurídica. Las abreviaturas de los nombres de autores, publicaciones periódicas u

ocasionales y obras autónomas figuran, de modo respectivo, en los *apéndices I, II y III* de este sitio web, estando en toda hipótesis de acuerdo fundamental con las más generalizadas normas y usos internacionales. En línea con la combinación o con su final, en el margen derecho, es de advertir que aparece de nuevo todo restrictivo —en tipo menos destacado y entre corchetes— con la tilde que indica la sílaba en que la intensidad tónica deberá, o al menos podrá, recaer correctamente, información que parece no poco útil; y, también "ad informationem", una mayúscula inicia esos restrictivos en el caso de que se los haya de considerar nombres en aposición, lo sean de lugares y personas o estén relacionados con éstas, aunque la vieja tradición botánico-tipográfica vaya cayendo en desuso no discutible. En el *Índice de nombres científicos* se dan, en letra negrita, los nombres de los táxones considerados correctos en los VIII primeros volúmenes de *Flora ibérica*.

2.º **Sinónimos.** En el cuerpo de la obra deberán ir los nombres no correctos que han figurado en las *Floras Básicas* en el rango específico y subespecífico, más algún otro que lo merezca por otra razón, a juicio de los editores. Cuando existe un basiónimo como tal —o nombre sustituido, en su caso—, va en primer término, con cita bibliográfica plena, y se lo señala expresamente. Se hizo un esfuerzo no pequeño para que nuestro índice recoja toda la sinonimia útil, incluso en el rango varietal. En el *Índice de nombres científicos* se dan, en letra redondita, los sinónimos aparecidos en los VIII primeros volúmenes de *Flora ibérica*.

3.º **Indicación locotípica.** Se transcribe literalmente la parte del protólogo que se refiere a lo geográfico y, cuando es necesario, se añade alguna palabra o frase complementaria o aclaratoria de aquélla.

4.º **Iconografía.** En este ítem pueden figurar dos tipos de referencias: las que remiten a icones, de cierta calidad, publicados en obras a poder ser accesibles; y las que remiten a nuestras láminas originales —una por género, al menos—, así como a las figuras explicativas. En concreto, el total de los endemismos ibéricos o baleares no iconografiados ya en forma satisfactoria y en obra fácilmente accesible, lo es aquí de manera prioritaria. En el *Índice de táxones iconografiados* se relacionan los táxones de los que se da un dibujo original en los VIII primeros volúmenes de *Flora ibérica*.

5.º **Descripción.** Se procuró que fuese concisa, y como ya está dicho, se ha sacrificado la homogeneidad en aras de la funcionalidad, aunque tal sacrificio implique



alguna falta de proporción externa. Se puso interés en que no fuese omitido ningún carácter diagnóstico de importancia.

6.º **Números cromosómicos.** Se los indica siempre que son conocidos y refiriéndonos a recuentos ibéricos, de modo preferencial: solo acudimos a recuentos llevados a cabo en plantas extraibéricas —a los que se añade un asterisco— en el caso de que no se conozca ninguno ibérico, más en el de que los ibéricos difieran de los extraibéricos. En el *Apéndice VII* de este sitio web se da una relación concisa del contenido de la base de datos de números cromosómicos de plantas vasculares ibéricas.

7.º **Hábitat.** Nos referimos a él de modo que la flora sea fácilmente inteligible a los no expertos en asunto conflictivo por demás. De forma concreta, deliberada, hemos renunciado al uso de la terminología fitosociológica, no solo por inconvenientes internos —falta de homogeneidad y estabilidad—, sino también por el hecho innegable de que hasta muchos taxónomos la desconocen.

8.º **Floración.** Nuestros datos relativos a la floración no han de ser tomados en sentido estricto, absoluto, ya que ésta depende muchísimo de factores múltiples, en territorio muy amplio y heterogéneo. Señalamos como topes los meses primero y último que indican las etiquetas de los pliegos útiles disponibles.

9.º **Distribución geográfica.** Se indica de modo muy esquemático la distribución total del taxon referido, para luego concretar la distribución peninsular y/o baleárica, en breve frase; detallándose, por fin, las provincias —tanto españolas como portuguesas— e islas en que su presencia consta por el material de los *Herbarios Básicos* o, en géneros sin graves problemas taxonómicos, por el testimonio en firme de los asesores y, de manera ocasional —siglas entre paréntesis—, por el testimonio de algún especialista, en géneros o grupos conflictivos, o por citas fiables aunque no respaldadas por pliego ninguno; los corchetes, a su vez, denotan que se trata de taxón meramente alóctono en la provincia, pero que, por uno u otro motivo, puede no parecerlo; en tanto que una interrogación se refiere a dudas geográficas o taxonómicas, leves de ordinario. Y el signo † indica extinción local de la planta. Con el signo l van señalados los táxones endémicos. Precede también, alfabéticamente, la indicación **And.** —Andorra—, en su caso. He aquí los códigos utilizados para España y Portugal, en orden siempre alfabético:

España (Esp.)					
<b>A</b>	Alicante	<b>Gu</b>	Guadalajara	<b>Mn</b>	Menorca
<b>Ab</b>	Albacete	<b>H</b>	Huelva	<b>Ib</b>	Ibiza
<b>Al</b>	Almería	<b>Hu</b>	Huesca	<b>Po</b>	Pontevedra
<b>Av</b>	Ávila	<b>J</b>	Jaén	<b>S</b>	Cantabria (Santander)
<b>B</b>	Barcelona	<b>L</b>	Lérida (Lleida)	<b>Sa</b>	Salamanca
<b>Ba</b>	Badajoz	<b>Le</b>	León	<b>Se</b>	Sevilla
<b>Bi</b>	Vizcaya	<b>Lo</b>	La Rioja (Logroño)	<b>Sg</b>	Segovia
<b>Bu</b>	Burgos	<b>Lu</b>	Lugo	<b>So</b>	Soria
<b>C</b>	La Coruña (A Coruña)	<b>M</b>	Madrid	<b>SS</b>	Guipúzcoa
<b>Ca</b>	Cádiz	<b>Ma</b>	Málaga	<b>T</b>	Tarragona
<b>Cc</b>	Cáceres	<b>Mu</b>	Murcia	<b>Te</b>	Teruel
<b>Co</b>	Córdoba	<b>Na</b>	Navarra	<b>To</b>	Toledo
<b>CR</b>	Ciudad Real	<b>O</b>	Asturias (Oviedo)	<b>V</b>	Valencia
<b>Cs</b>	Castellón	<b>Or</b>	Orense (Ourense)	<b>Va</b>	Valladolid
<b>Cu</b>	Cuenca	<b>P</b>	Palencia	<b>Vi</b>	Álava
<b>Ge</b>	Gerona (Girona)	<b>PM</b>	Islas Baleares	<b>Z</b>	Zaragoza
<b>Gr</b>	Granada	<b>Mll</b>	Mallorca	<b>Za</b>	Zamora

Portugal (Port.)			
<b>AAI</b>	Alto Alentejo	<b>DL</b>	Douro Litoral
<b>Ag</b>	Algarve	<b>E</b>	Estremadura
<b>BA</b>	Beira Alta	<b>Mi</b>	Minho
<b>BAI</b>	Baixo Alentejo	<b>R</b>	Ribatejo
<b>BB</b>	Beira Baixa	<b>TM</b>	Trás-os-Montes
<b>BL</b>	Beira Litoral		



10. **Nombres vernáculos.** A continuación, si ha lugar, enumeramos los nombres peninsulares o insulares que más comúnmente se aplican a la planta, entendiéndose por nombre vernáculo no solo el vulgar o usado por el vulgo, sino cualquier otro admitido, por una u otra vía, en los idiomas que nos conciernen, Quede claro que no aspiramos a ser exhaustivos en este punto y que apenas hemos hecho investigación propia: nuestra

información es, fundamentalmente, bibliográfica. Y como quiera que de un solo nombre puede haber multitud de variantes locales, explicitemos que habrá de atenderse, dentro de cada grupo, a las más extendidas. Como es lógico, se hace constar si los nombres en cuestión corresponden a los idiomas castellano —sobreentendido—, catalán, euskera, gallego y portugués, del continente o no, sin excesivas distinciones. Bien sería que nuestra flora —y no solo en este dominio— diese pie a síntesis o recopilaciones más perfectas, plenamente investigatorias en su prometedor alcance. En el *Índice de nombres vernáculos* se relacionan los aparecidos en los VIII primeros volúmenes de *Flora ibérica*.

11. **Observaciones.** Con frecuencia, en párrafo autónomo, solemos añadir sucinta información sobre propiedades y demás. Ocasionalmente preceden o siguen algunas observaciones, de índole sobre todo taxonómica: en ellas nos hemos referido más de una vez a presuntas variedades, rango éste cuyo tratamiento es aquí heterogéneo, lo que se imponía señalar.

12. **Especies que han de buscarse.** Indicamos en este ítem aquellas de presencia probable en nuestro territorio, aunque no confirmada.

13. **Los híbridos,** cuya existencia en el territorio de la Flora consta en firme, simplemente se indican tras cada género.

En esta misma página se puede encontrar una determinación de las plantas tanto en familias como en géneros.

Esta pantalla muestra la determinación de las plantas con respecto a la división por familias:

**Lista alfabética de Familias:**

A-C	C-L	L-P	P-Z
<a href="#">Acanthaceae</a> (14)	<a href="#">Cornaceae</a> (8)	<a href="#">Lemnaceae</a> (18)	<a href="#">Psilotaceae</a> (1)
<a href="#">Aceraceae</a> (9)	<a href="#">Crassulaceae</a> (5)	<a href="#">Lentibulariaceae</a> (14)	<a href="#">Pteridaceae</a> (1)
<a href="#">Adiantaceae</a> (1)	<a href="#">Cruciferae</a> (4)	<a href="#">Lilaeaceae</a> (17)	<a href="#">Punicaceae</a> (8)
<a href="#">Adoxaceae</a> (15)	<a href="#">Cryptogrammaceae</a> (1)	<a href="#">Liliaceae</a> (20)	<a href="#">Pyrolaceae</a> (4)
<a href="#">Agavaceae</a> (20)	<a href="#">Cucurbitaceae</a> (3)	<a href="#">Linaceae</a> (9)	<a href="#">Rafflesiaceae</a> (8)
<a href="#">Aizoaceae</a> (2)	<a href="#">Culcitaceae</a> (1)	<a href="#">Lycopodiaceae</a> (1)	<a href="#">Ranunculaceae</a> (1)
<a href="#">Alismataceae</a> (17)	<a href="#">Cupressaceae</a> (1)	<a href="#">Lythraceae</a> (8)	<a href="#">Resedaceae</a> (4)
<a href="#">Amaranthaceae</a> (2)	<a href="#">Cynomoriaceae</a> (8)	<a href="#">Malvaceae</a> (3)	<a href="#">Rhamnaceae</a> (9)
<a href="#">Amaryllidaceae</a> (20)	<a href="#">Cyperaceae</a> (18)	<a href="#">Marsileaceae</a> (1)	<a href="#">Rosaceae</a> (6)
<a href="#">Anacardiaceae</a> (9)	<a href="#">Chenopodiaceae</a> (2)	<a href="#">Martyniaceae</a> (14)	<a href="#">Rubiaceae</a> (15)
<a href="#">Apocynaceae</a> (11)	<a href="#">Davalliaceae</a> (1)	<a href="#">Meliaceae</a> (9)	<a href="#">Ruppiaceae</a> (17)
<a href="#">Aquifoliaceae</a> (8)	<a href="#">Dioscoreaceae</a> (21)	<a href="#">Menyanthaceae</a> (11)	<a href="#">Rutaceae</a> (9)
<a href="#">Araceae</a> (18)	<a href="#">Dipsacaceae</a> (15)	<a href="#">Molluginaceae</a> (2)	<a href="#">Salicaceae</a> (3)
<a href="#">Araliaceae</a> (10)	<a href="#">Droseraceae</a> (5)	<a href="#">Monotropaceae</a> (4)	<a href="#">Salvinaceae</a> (1)
<a href="#">Aristolochiaceae</a> (1)	<a href="#">Ebenaceae</a> (5)	<a href="#">Moraceae</a> (3)	<a href="#">Santalaceae</a> (8)
<a href="#">Asclepiadaceae</a> (11)	<a href="#">Elaeagnaceae</a> (8)	<a href="#">Musaceae</a> (18)	<a href="#">Sapindaceae</a> (9)
<a href="#">Aspidiaceae</a> (1)	<a href="#">Elatinaceae</a> (3)	<a href="#">Myoporaceae</a> (14)	<a href="#">Sapotaceae</a> (5)
<a href="#">Aspleniaceae</a> (1)	<a href="#">Empetraceae</a> (4)	<a href="#">Myricaceae</a> (2)	<a href="#">Saxifragaceae</a> (5)
<a href="#">Athyriaceae</a> (1)	<a href="#">Ephedraceae</a> (1)	<a href="#">Myrtaceae</a> (8)	<a href="#">Scrophulariaceae</a> (13)
<a href="#">Azollaceae</a> (1)	<a href="#">Equisetaceae</a> (1)	<a href="#">Najadaceae</a> (17)	<a href="#">Selaginellaceae</a> (1)
<a href="#">Balsaminaceae</a> (9)	<a href="#">Ericaceae</a> (4)	<a href="#">Nyctaginaceae</a> (2)	<a href="#">Simaroubaceae</a> (9)
<a href="#">Basellaceae</a> (2)	<a href="#">Euphorbiaceae</a> (8)	<a href="#">Nymphaeaceae</a> (1)	<a href="#">Sinopteridaceae</a> (1)
<a href="#">Berberidaceae</a> (1)	<a href="#">Fagaceae</a> (2)	<a href="#">Oleaceae</a> (11)	<a href="#">Smilacaceae</a> (21)
<a href="#">Betulaceae</a> (2)	<a href="#">Frankeniaceae</a> (3)	<a href="#">Onagraceae</a> (8)	<a href="#">Solanaceae</a> (11)
<a href="#">Bignoniaceae</a> (14)	<a href="#">Gentianaceae</a> (11)	<a href="#">Ophioglossaceae</a> (1)	<a href="#">Sparganiaceae</a> (18)
<a href="#">Blechnaceae</a> (1)	<a href="#">Geraniaceae</a> (9)	<a href="#">Orchidaceae</a> (21)	<a href="#">Tamaricaceae</a> (3)
<a href="#">Boraginaceae</a> (11)	<a href="#">Gesneriaceae</a> (14)	<a href="#">Orobanchaceae</a> (14)	<a href="#">Taxaceae</a> (1)
<a href="#">Botrychiaceae</a> (1)	<a href="#">Globulariaceae</a> (14)	<a href="#">Osmundaceae</a> (1)	<a href="#">Theligonaceae</a> (8)
<a href="#">Buddleiaceae</a> (13)	<a href="#">Gramineae</a> (19)	<a href="#">Oxalidaceae</a> (9)	<a href="#">Thelypteridaceae</a> (1)
<a href="#">Butomaceae</a> (17)	<a href="#">Grossulariaceae</a> (5)	<a href="#">Paeoniaceae</a> (3)	<a href="#">Thymelaeaceae</a> (8)
<a href="#">Buxaceae</a> (8)	<a href="#">Guttiferae</a> (3)	<a href="#">Palmae</a> (18)	<a href="#">Tiliaceae</a> (3)
<a href="#">Cactaceae</a> (2)	<a href="#">Haloragaceae</a> (8)	<a href="#">Papaveraceae</a> (1)	<a href="#">Trapaceae</a> (8)
<a href="#">Callitrichaceae</a> (12)	<a href="#">Hemionitidaceae</a> (1)		<a href="#">Tropaeolaceae</a> (9)
<a href="#">Campanulaceae</a> (14)	<a href="#">Hippocastanaceae</a> (9)	<a href="#">Phytolaccaceae</a> (2)	<a href="#">Typhaceae</a> (18)
<a href="#">Cannabaceae</a> (3)	<a href="#">Hippuridaceae</a> (8)	<a href="#">Pinaceae</a> (1)	<a href="#">Ulmaceae</a> (3)
<a href="#">Cannaceae</a> (18)	<a href="#">Hydrangeaceae</a> (5)	<a href="#">Pittosporaceae</a> (5)	<a href="#">Umbelliferae</a> (10)
<a href="#">Capparaceae</a> (3)	<a href="#">Hydrocharitaceae</a> (17)	<a href="#">Plantaginaceae</a> (13)	<a href="#">Urticaceae</a> (3)
<a href="#">Caprifoliaceae</a> (15)	<a href="#">Hydrophyllaceae</a> (11)	<a href="#">Platanaceae</a> (2)	<a href="#">Valerianaceae</a> (15)
<a href="#">Caryophyllaceae</a> (2)	<a href="#">Hymenophyllaceae</a> (1)	<a href="#">Plumbaginaceae</a> (2)	<a href="#">Verbenaceae</a> (12)
<a href="#">Casuarinaceae</a> (2)	<a href="#">Hypolepidaceae</a> (1)	<a href="#">Polygalaceae</a> (9)	<a href="#">Violaceae</a> (3)
<a href="#">Celastraceae</a> (8)	<a href="#">Iridaceae</a> (20)	<a href="#">Polygonaceae</a> (2)	<a href="#">Viscaceae</a> (8)
<a href="#">Ceratophyllaceae</a> (1)	<a href="#">Isoetaceae</a> (1)	<a href="#">Polypodiaceae</a> (1)	<a href="#">Vitaceae</a> (9)
<a href="#">Cistaceae</a> (3)	<a href="#">Juglandaceae</a> (9)	<a href="#">Pontederiaceae</a> (18)	<a href="#">Woodsiaceae</a> (1)
<a href="#">Cneoraceae</a> (9)	<a href="#">Juncaceae</a> (17)	<a href="#">Portulacaceae</a> (2)	<a href="#">Zannichelliaceae</a> (17)
<a href="#">Commelinaceae</a> (17)	<a href="#">Juncaginaceae</a> (17)	<a href="#">Posidoniaceae</a> (17)	<a href="#">Zosteraceae</a> (17)
<a href="#">Compositae</a> (16)	<a href="#">Labiales</a> (12)	<a href="#">Potamogetonaceae</a> (17)	<a href="#">Zygophyllaceae</a> (9)
<a href="#">Convolvulaceae</a> (11)	<a href="#">Lauraceae</a> (1)	<a href="#">Primulaceae</a> (5)	
<a href="#">Coriariaceae</a> (9)	<a href="#">Leguminosae</a> (7)	<a href="#">Proteaceae</a> (8)	

En esta determinación se pueden ver todos los nombres de las familias de las que consta la flora ibérica en orden alfabético.

Aquí podemos observar la enorme variedad de familias que posee la flora de nuestra península y las islas Baleares, ya que poseemos ver más de 100 familias distintas y cada una de ellas con un mínimo de 2 plantas.

De cada una de ellas se ofrece la información que describimos anteriormente, aunque aun existen familias de las que su información no esta disponible a esta fecha ya que dicha determinación esta en continuo cambio debido a la aparición de nuevas especies.

Si se pulsa en las los hipervínculos, en las familias que están en azul, te aparece su descripción así como cada una de las plantas de las que esta compuesta dicha familia.

Un ejemplo es el siguiente:

**Géneros aceptados de la Familia *Acanthaceae***  
● Descripción de la Familia

● **Acanthus** ● Nomenclatura ● Descripción (pdf)

Atrás

Si pulsas en descripción de la familia aparece:

#### CLI. ACANTHACEAE \*

Hierbas anuales o perennes –arbustos o raramente árboles en especies extraibéricas–, generalmente con pelos, glandulíferos o no glandulíferos. Hojas opuestas –excepto en las *Nelsonioideae* Lindl. ex Pfeiff.–, simples, sésiles, a menudo con cistolitos, sin estípulas. Flores hermafroditas, irregulares, solitarias o en cimas racemiformes, con brácteas y bractéolas en la base –a menudo grandes y petaloideas–, que frecuentemente las envuelven. Cáliz con (3)4-5 sépalos soldados, lobulado o hendido, bilabiado o acampanado. Corola generalmente bilabiada y con 5 lóbulos –a veces falta el labio superior–. Estambres 2 ó 4(5), didínamos, alternipétalos, insertos en el tubo de la corola, reducidos algunos, a veces, a estaminodios; anteras bitecas –a veces, una de las tecas mucho más grande que la otra–, o monotecas por reducción, con dehiscencia longitudinal. Ovario súpero, bicarpelar, con 2 lóculos, de placentación axilar y, generalmente, con 2 rudimentos seminales; estilo simple, largo; estigma bilobulado. Fruto en cápsula con dehiscencia explosiva, loculicida por 2 valvas. Semillas sin endosperma, dispuestas sobre funículos ganchudos que facilitan su diseminación y de testa, en algunas especies extraibéricas, cubierta de pelos o escamas que se hacen pegajosos o viscosos al humedecerse.

Comprende unos 250 géneros y c. 2500 especies, que se distribuyen principalmente por las regiones tropicales.

*Observaciones.*—Algunas especies de un cierto número de géneros se cultivan

Si pulsas en nomenclatura:

**Nombres aceptados de *Acanthus***  
(Cualquier vínculo de nombre aceptado, le mostrará los sinónimos del mismo)

[Acanthus](#) L., Sp. Pl. 639 ["939"] (1753)  
[Acanthus mollis](#) L., Sp. Pl. 639 ["939"] (1753)

Atrás

Si pulsas en descripción (pdf):

78      CLI. ACANTHACEAE  
1. **Acanthus**

### 1. *Acanthus* L.\*

[Acánthus, -i m. – gr. *ákanthos*, -ou m.f.; lat. *acanthus*(-os), -i m.f. = acanto, principalmente, dos especies del género *Acanthus* L. (Acantáceas), el *A. mollis* L. y el *A. spinosus* L.; aunque también diversas especies espinosas del género *Acacia* Mill. (Leguminosas), de Arabia y Egipto –gr. *ákantha*, -ēs f. = espina, pincho, etc.]

Hierbas perennes –o pequeños arbustos en especies extraibéricas–. Tallo simple, erecto, escaposo. Hojas en su mayoría basales, pinnatipartidas, pinnatífidas o pinnatisectas, sin cistolitos. Flores en espigas terminales, densas, con una bráctea foliácea, espinosa y, generalmente, con 2 bractéolas más pequeñas y enteras. Cáliz con 4 sépalos, los 2 laterales (internos) mucho más cortos que los otros 2, que son foliáceos. Corola con un solo labio –el inferior–, trilobulado; tubo muy corto. Estambres 4, didínamos, más cortos que el labio de la corola; anteras monotecas, ciliadas. Ovario ovoide, bilocular, con 2 rudimentos seminales en cada lóculo; estilo largo, con 2 ramas estigmatíferas cortas. Fruto en cápsula, ovoide, comprimida, coriácea. Semillas 2-4, comprimidas, lisas o tuberculado-rugosas, glabras.

*Bibliografía.*–M. Rix in *Plantsman* 2(3): 132-140 (1981).

El numero de plantas por el que esta compuesta cada familia aparece entre paréntesis después del nombre de las familias.

Las familias que están subrayadas aun faltan por generar la documentación de su descripción así como la de las plantas que la integran.



Otra posible determinación es por géneros. En esta otra imagen se puede ver la determinación de este tipo:

**Lista alfabética de géneros:**

En letra negrita los géneros publicados; en subrayado, las familias en preparación; y entre paréntesis, el volumen.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
•	Bacharis, Compositae (16)																									
•	Bacopa, <u>Scrophulariaceae</u> (13)																									
•	Baldellia, Alismataceae (17)																									
•	Ballota, <u>Labiatae</u> (12)																									
•	Balsamita, Compositae (16)																									
•	<b>Barbarea</b> , Cruciferae (4)																									
•	<b>Barlia</b> , Orchidaceae (21)																									
•	Bartsia, <u>Scrophulariaceae</u> (13)																									
•	<b>Bassia</b> , Chenopodiaceae (2)																									
•	Bellardiochloa, Gramineae (19)																									
•	Bellevia, Liliaceae (20)																									
•	Bellis, Compositae (16)																									
•	Bellium, Compositae (16)																									
•	<b>Berberis</b> , Berberidaceae (1)																									
•	<b>Bergia</b> , Elatinaceae (3)																									
•	<b>Berula</b> , Umbelliferae (10)																									
•	<b>Beta</b> , Chenopodiaceae (2)																									
•	<b>Betula</b> , Betulaceae (2)																									
•	<b>Biarum</b> , Araceae (18)																									
•	Bidens, Compositae (16)																									
•	<b>Bifora</b> , Umbelliferae (10)																									
•	<b>Biscutella</b> , Cruciferae (4)																									
•	<b>Bituminaria</b> , Leguminosae (7)																									
•	Blackstonia, Gentianaceae (11)																									
•	<b>Blechnum</b> , Blechnaceae (1)																									
•	<b>Blymus</b> , Cyperaceae (18)																									
•	Blyxa, Hydrocharitaceae (17)																									
•	<b>Bolboschoenus</b> , Cyperaceae (18)																									
•	<b>Boleum</b> , Cruciferae (4)																									
•	Bombycilaena, Compositae (16)																									
•	Borago, Boraginaceae (11)																									
•	<b>Borderea</b> , Dioscoreaceae (21)																									
•	<b>Boreava</b> , Cruciferae (4)																									
•	Borrichia, Compositae (16)																									
•	<b>Botrychium</b> , Botrychiaceae (1)																									
•	<b>Boussingaultia</b> , Basellaceae (2)																									
•	Bouteloua, Gramineae (19)																									
•	<b>Bowlesia</b> , Umbelliferae (10)																									
•	Bracharia, Gramineae (19)																									
•	Brachypodium, Gramineae (19)																									
•	<b>Brassica</b> , Cruciferae (4)																									
•	Brimeura, Liliaceae (20)																									
•	Briza, Gramineae (19)																									
•	Bromus, Gramineae (19)																									
•	<b>Bryonia</b> , Cucurbitaceae (3)																									
•	Buddleja, <u>Buddlejaceae</u> (13)																									
•	<b>Bufonia</b> , Caryophyllaceae (2)																									
•	Buglossoides, Boraginaceae (11)																									
•	Bulbocodium, Liliaceae (20)																									
•	<b>Bulbostylis</b> , Cyperaceae (18)																									
•	<b>Bunias</b> , Cruciferae (4)																									
•	<b>Bunium</b> , Umbelliferae (10)																									
•	<b>Bupleurum</b> , Umbelliferae (10)																									
•	Butomus, Butomaceae (17)																									
•	<b>Buxus</b> , Buxaceae (8)																									

En la determinación por géneros aparece cada una de las plantas y a su derecha el nombre de la familia a la que pertenece. Cuando queramos conocer sus características solo debemos de pinchar en el nombre de la familia y nos aparecerá la descripción de dicha familia.



En el vínculo que pone a su derecha Nomenclatura nos aparece

**Nombres aceptados de *Abies***

(Cualquier vínculo de nombre aceptado, le mostrará los sinónimos del mismo)

[Abies](#) Mill., Gard. Dict. Abr. ed. 4 (1754)

[Abies alba](#) Mill., Gard. Dict. ed. 8 n.º 1 (1768)

[Abies pinsapo](#) Boiss., Notice Abies Pinsapo 8 (1838)

[Atrás](#)

En el vínculo que pone Descripción (pdf) nos aparece una descripción en formato pdf. Este formato quiere decir que se puede ver teniendo en el ordenador que este utilizando el programa Acrobat Reader.

### 1. *Abies* Mill.\*

[Ábies f. – lat. *abies* = abeto]

Árboles de hoja perenne, con ramas verticiladas regularmente, que forman copa cónica o piramidal. Ramillas redondeadas, sin surcos; cicatriz de las hojas circular. Hojas lineares, aplanadas o raramente subtetrales, con 2 bandas estomáticas en la cara inferior; 2 canales resiníferos, marginales o centrales. Conos masculinos axilares y numerosos en la cara inferior de las ramas. Estróbilos erectos, de cilíndricos a ovoides, situados solo en la parte superior de la copa; escamas caducas, desarticulándose en el otoño del primer año del raquis leñoso, persistente. Semillas aladas.

*Observaciones.*—Las hojas de las ramas superiores de la copa son frecuentemente muy diversas de las situadas en las ramas inferiores. En parques y jardines se cultivan a veces especies exóticas no mencionadas en esta Flora.

*Bibliografía.*—J. DO AMARAL FRANCO, Abetos (1950).

1. Hojas discolores, flexibles, subdísticas en las ramas inferiores sombreadas; estróbilo con las escamas tectrices exertas ..... 1. *A. alba*
- Hojas concoloras, rígidas, en disposición radiada; estróbilo con las escamas tectrices inclusas ..... 2. *A. pinsapo*

En el vínculo que pone Imprenta (pdf), es un documento que puede ser impreso.

**Anarrhinum** Desf. [nom. cons.]\*

[etimología

c.

3-4

líneas]

Hierbas bienales o vivaces, en ocasiones algo leñosas en la base, glabras, glandular-pubescentes o papilosas, con indumento homótrico formado por papilas o pelos glandulíferos, más raramente arbustos ± leñosos, glabros. Tallos homomorfos, erectos, simples o escasamente ramificados. Hojas palmatinervias, heteromorfas, raramente ± homomorfas, en general superiores e inferiores simultáneamente en la planta; las inferiores en roseta basal, ± espatuladas, dentadas, serradas o crenadas, obtusas, estrechadas hacia la base, planas, pecioladas; las superiores alternas, enteras, entonces de lineares a lanceoladas o, con más frecuencia, 3-7 palmatinervias —con lóbulos generalmente desiguales, enteros, dentados o lobulados, de lineares a suborbiculares— agudas u obtusas, estrechadas o no hacia la base, planas, sésiles o brevemente pecioladas; pecíolo, en su caso, más corto que el limbo, no cirroso. Inflorescencia en largo racimo o panícula terminal, densa, de eje recto, con brácteas similares a las hojas, progresivamente reducidas. Flores zigomorfas, de pediceladas a subsésiles, solitarias en la axila de las brácteas; pedicelos ± largos, no acrescentes, erectos en fruto. Cáliz profundamente hendido, con 5 sépalos, subiguales, ± unidos, no imbricados, más

En el vínculo que pone Borrador (pdf) nos aparece un borrador de la familia a la que nos referimos. Aun el documento no esta completo o revisado.

**2. Ajuga L.\*** \*F. Llamas 24-I-2008

[Ajuga, n. – gr. ágyios = débil de miembros, ya que era utilizada contra la gota. Scribunius Largus lo considera derivado del latín agibo = echar fuera, al ser plantas utilizadas como abortivo]

Plantas herbáceas, anuales o perennes, rara vez leñosas en la base, glabras o pelosas. Hojas enteras o crenadas, elípticas, ovadas, espatuladas, o trifidas, con segmentos lineares. Inflorescencia espiciforme, ± densa, formada por verticilastros. Brácteas en general como las hojas y sobresaliendo de los verticilastros. Cáliz campanulado, casi actinomórfico, con 5 dientes casi iguales, de lineares a triangulares. Corola con tubo que sobresale del cáliz, con un anillo de pelos por dentro; labio superior muy corto, rudimentario emarginado; labio inferior trilobulado, con el lóbulo central mayor, ± dividido, dirigido hacia abajo, algo cóncavo. Estambres 4, didinamos, en general exsertos del tubo. Estilo no ginobásico; estigma bilobulado. Núculas ± obovoides, reticuladas.

Se compone de unas 40 especies, distribuidas por toda Eurasia.

1. Hojas trifidas, con tres segmentos estrechos 1. *A. chamaepitys*
- Hojas enteras, dentadas o lobuladas 2
2. Hojas estrechas, 3-6 mm de ancho, de lineares a oblongas 2. *A. iva*
- Hojas anchas, 8-50 mm 3
3. Planta con estolones aéreos; tallos pelosos en dos caras opuestas; estambres pelosos 3. *A. reptans*

En esta determinación vemos lo difícil que resulta dar una descripción de las familias con respecto a sus características así como las características de cada una de las plantas que

pertenecen a ellas, debido a que la gran mayoría de estas son términos científicos que solo un experto en la materia podría conocer.

Por lo tanto si una persona con desconocimiento en la materia quisiese conocer a que familia pertenece la planta que desea determinar, le resultaría difícil, por no decir imposible, ya que por un lado tendría que conocer los términos científicos de las características que posee esa planta, y si esa persona no le resultan conocidos además de esta dificultad tendríamos que añadir, lo complicado que resulta encontrar la familia a la que pertenece una planta teniendo que leerse todo el documento sobre el que estamos hablando.

Tendría que aprender las características de la planta sobre la que quiere realizar su determinación, con lo que debe de conocer todas las características por las que se puede describir una planta. Debe de realizar un estudio de las características que puede poseer una planta así como la aplicación de estas a su planta a determinar.

Además de esto, debe de ir mirando en todas las familias de las más de 100 que existen en la flora ibérica, una por una y mirando si coinciden las características de su planta con las características que se describen en las familias. Esto resultaría una tarea costosa, difícil y llevaría mucho tiempo su realización. Ya que además de comparar las características de su planta con las de cada familia, puede que alguna de estas características tenga varias posibilidades para una misma familia por lo que puede que esa planta tenga un grado de pertenencia según las características de las que el usuario tenga información.

Por este motivo nosotras vamos a realizar inicialmente una herramienta nítida, para que el usuario vea que según las características que introduzca la familia o familias que pueden ser aceptadas con las características que el usuario ha introducido.

Seguidamente damos otra herramienta borrosa para que el usuario compruebe que en el caso de que varias familias estén aceptadas, el grado de pertenencia a ellas introduciendo dos nuevas características, en este caso dos elementos de carácter borroso. Con este grado de pertenencia podemos comprobar cual es la familia que mejor describe las características que el usuario ha introducido en la herramienta, siendo esta la que mayor grado de pertenencia obtenga.

Después el biólogo nos proporcionó una serie de características para realizar una herramienta que le resultase sencilla de utilizar y que tuviese las características que ellos utilizan para determinar las familias de plantas. Por ello realizaremos una ultima herramienta para finalizar nuestro proyecto.

## SEGUNDA DETERMINACIÓN

Otra forma de determinación es la que se muestra en la siguiente página:

<http://www.floradeiberia.com/clasificación.php>

En esta determinación primero se separan los reinos de las algas y del reino de los hongos. La composición de esto se denomina líquenes.

Luego sigue con el reino de las plantas que es que a nosotros nos incumbe en este momento.

Este reino esta subdivido en las plantas gimnospermas y angiospermas. De estos términos hemos dado su descripción anteriormente.

En esta determinación se muestran todas sus familias pero en ellas no se puede comprobar cada una de las características que posee cada familia.

Por este motivo esta determinación no resulta de gran ayuda a la hora de darle al usuario una respuesta cuando este quiere conocer la familia a la que pertenece una planta.

Como en esta página no se proporciona descripción alguna, no se puede conocer las características que posee la familia con lo que aun sabiendo las características de la planta, no se podría conocer si dicha familia posee las mismas o incluso similares.

Por lo tanto esta pagina no resulta de ninguna ayuda para conocer la similitud entre la planta a determinar y la familia a la que pertenece.

La determinación que realizan es la siguiente:

En verde se pueden ver el nombre de las distintas familias de la flora ibérica.

## Reino Protocista (Algas)

## Reino Fungi (Hongos)

## Liquenes (Alga + Hongo)

## Reino Plantae:

### División Briophyta:

Clase Bryopsida

Clase Hepaticopsida

Clase Anthocerotopsida

### División Pteridophyta:

### División Spermatophyta:

GIMNOSPERMAE - GIMNOSPERMAS.

-Subdivisión Cycadophytina

-Clase Cycadopsida

-Orden Cycadales

-Familia Cycadaceae

-Subdivisión Coniferophytina

-Clase Ginkgopsida

-Orden Ginkgoales

-Familia Ginkgoaceae

-Clase Coniferopsida

-Orden Coniferales

-Familia Pinaceae

-Familia Cupressaceae

-Familia Taxodiaceae

-Familia Araucariaceae

-Familia Taxaceae

-Familia Cephalotaxaceae

-Familia Podocarpaceae

-Subdivisión Gnetales

-Clase Gnetales

-Orden Gnetales

-Familia Ephedraceae

ANGIOSPERMAE - ANGIOSPERMAS

-Subdivisión Magnoliophytina

-Clase Magnoliopsida

-Subclase Liliidae - monocotiledoneas

-Superorden Liliales

-Orden Liliales

- Orden Ranunculales
  - Familia Berberidaceae
  - Familia Ranunculaceae
- Superorden Proteanae
  - Orden Proteales
    - Familia Proteaceae
    - Familia Platanaceae
  - Orden Buxales
    - Familia Buxaceae
- Superorden Saxifraganae
  - Orden Saxifragales
  - Orden Poales
    - Familia Gramineae
  - Orden Zingiberales
    - Familia Musaceae
- Subclase Magnoliidae
  - Superorden Magnoliana
    - Orden Magnoliales
      - Familia Magnoliaceae
      - Familia Annonaceae
    - Orden Laurales
      - Familia Lauraceae
- Subclase Ranunculidae
  - Orden Ranunculales
    - Familia Berberidaceae
    - Familia Ranunculaceae
- Superorden Proteanae
  - Orden Proteales
    - Familia Proteaceae
    - Familia Platanaceae
  - Orden Buxales
    - Familia Buxaceae
- Superorden Saxifraganae
  - Orden Saxifragales

- Familia Grossulariaceae
- Familia Altingiaceae
- Superorden Rosanae
  - Orden Vitales
    - Familia Vitaceae
  - Orden Malpighiales
    - Familia Linaceae
    - Familia Euphorbiaceae
    - Familia Guttiferae
    - Familia Salicaceae
  - Orden Celastrales
    - Familia Celastraceae
  - Orden Fabales
    - Familia Leguminosae
    - Familia Polygalaceae
  - Orden Fagales
    - Familia Betulaceae
    - Familia Fagaceae
    - Familia Juglandaceae
    - Familia Myricaceae
    - Familia Casuarinaceae
  - Orden Cucurbitales
    - Familia Coriariaceae
  - Orden Rosales
    - Familia Rosaceae
    - Familia Rhamnaceae
    - Familia Eleagnaceae
    - Familia Ulmaceae
    - Familia Moraceae
- Superorden Rutanae
  - Orden Myrtales
    - Familia Myrtaceae
    - Familia Lythraceae
  - Orden Malvales
    - Familia Cistaceae

- Familia Malvaceae
- Familia Bombacaceae
- Familia Sterculiaceae
- Familia Tiliaceae
- Familia Thymelaeaceae
- Orden Sapindales
  - Familia Rutaceae
  - Familia Cneoraceae
  - Familia Simaroubaceae
  - Familia Meliaceae
  - Familia Anacardiaceae
  - Familia Sapindaceae
  - Familia Hippocastaneaceae
  - Familia Aceraceae
- Orden Brassicales
  - Familia Capparaceae
  - Familia Cruciferae
- Superorden Santalanae
  - Orden Santalales
    - Familia Santalaceae
- Superorden Caryophyllanae
  - Orden Caryophyllales
    - Familia Chenopodiaceae
    - Familia Cactaceae
    - Familia Nyctaginaceae
    - Familia Phytolaccaceae
  - Orden Polygonales
    - Familia Tamaricaceae
    - Familia Plumbaginaceae
- Superorden Cornanae
  - Orden Cornales
    - Familia Cornaceae
    - Familia Hidrangeaceae
  - Orden Ericales



- Familia Ebenaceae
- Familia Sapotaceae
- Familia Theaceae
- Familia Ericaceae
- Familia Empetraceae
- Superorden Lamianae
  - Orden Gentianales
    - Familia Apocynaceae
    - Familia Asclepiadaceae
  - Orden Solanales
    - Familia Solanaceae
  - Orden Boraginales
    - Familia Boraginaceae
    - Familia Hydrophyllaceae
  - Orden Lamiales
    - Familia Oleaceae
    - Familia Scrophulariaceae
    - Familia Myoporaceae
    - Familia Globulariaceae
    - Familia Verbenaceae
    - Familia Buddlejaceae
    - Familia Bienoniaceae
    - Familia Labiatae
- Superorden Asteranae
  - Orden Aquifoliales
    - Familia Aquifoliaceae
    - Familia Aucubaceae
  - Orden Escalloniaceae
    - Familia Escalloniaceae
  - Orden Dipsacales
    - Familia Caprifoliaceae
  - Orden Apiales
    - Familia Araliaceae



### TERCERA DETERMINACIÓN

Otra forma de determinación es la que en un primer momento nos mostró nuestro director de proyecto que esta elaborada por los biólogos.

Aquí mostramos la pantalla principal de este documento que se encuentra en esta pagina web:

<http://www.rjb.csic.es/floraiberica/floraiberica/texto/pdfs/000%20clavegeneral.pdf>

Esta pagina nos la mostró nuestro codirector de proyecto Alfonso Garmendia Salvador profesor perteneciente al Departamento de Ecosistemas Agroforestales de la escuela Técnica Superior del Medio Rural y Enología, de la Universidad Politécnica de Valencia.

## CLAVE GENERAL\*

1. Plantas sin flores ni semillas, con esporas contenidas en esporangios,  $\pm$  agrupados en las axilas o en el envés de las hojas, o en conos terminales ..... **Pteridophyta** (véase vol. 1)
- Plantas con flores o conos —a veces muy simplificados— que llevan los rudimentos seminales o los sacos polínicos; semillas contenidas en frutos, en estróbilos o en gálbulos, raramente solitarias ..... 2
2. Flores unisexuales, generalmente agrupadas en conos masculinos y femeninos, las femeninas sin ovario, ni estilo, ni estigma; rudimentos seminales al descubierto; semillas solitarias o más frecuentemente contenidas en estróbilos o gálbulos; plantas leñosas, a menudo resinosas ..... **Gymnospermae** (véase vol. 1)
- Flores unisexuales o hermafroditas, solitarias o agrupadas en inflorescencias variadas, con ovario o estambres, o ambas cosas a la vez; rudimentos seminales encerrados en el ovario; semillas contenidas en frutos variados; plantas herbáceas o leñosas, generalmente no resinosas ..... **Angiospermae**

### Angiospermae

Incluye todas las familias de los volúmenes I al VIII, más el X, XIV, XV, XVIII y XXI, que van numeradas, y la gran mayoría de las familias de los demás volúmenes, aunque puede haber sido omitida alguna de las tan solo representadas por plantas introducidas.

1. Plantas acuáticas, sumergidas o flotantes, que no enraízan en el substrato ..... 2
- Plantas terrestres o acuáticas que enraízan en el substrato ..... 7
- 2[1]. Plantas sin hojas ni tallos claramente diferenciados ... **CLXXX. Lemnaceae** (vol. 18)
- Plantas con hojas y tallos claramente diferenciados ..... 3
- 3[2]. Hojas divididas en numerosos segmentos filiformes ..... 4
- Hojas no divididas en numerosos segmentos filiformes ..... 5
- 4[3]. Plantas con pequeñas vejigas en las hojas, o tallos aparentemente afilos ..... **CLII. Lentibulariaceae** p.p. (vol. 14)

Como se puede observar va siguiendo un método dicotómico en el que solo se puede elegir entre dos posibilidades, a continuación explicamos más detenidamente el método dicotómico.

A la hora de usar este documento para determinar las distintas plantas que los biólogos deseen, nos dimos cuenta que podía surgir un problema.

Este documento usa una búsqueda dicotómica.

La palabra DICOTÓMICA: DICO: Dos y TÓMICO: Partes, es decir se refiere a la DIVISIÓN en 2 partes. Puede tratarse de la Bifurcación de un tallo o de una rama. O también al Método de determinación en Claves Dicotómicas que dan 2 alternativas para una determinación.

**DICOTÓMICO** es la División o bifurcación de un eje en dos ramas más o menos iguales, tratándose de tallos o ramas.

**CLAVE DICOTÓMICA:** Se denomina dicotómica porque las características se presentan de 2 en 2, de manera que si tú eliges la primera, automáticamente queda descartada la segunda. Al haber elegido una u otra opción, te irá mandando que elijas entre más opciones, que tú irás sabiendo, observando al animal o a la planta.

O sea se presentan 2 alternativas y luego de acuerdo a las características que presenten, se elige la que corresponda a la especie que estás estudiando o analizando.

Según la búsqueda dicotómica la persona que realice dicha búsqueda va a ir respondiendo una serie de características con dos o varias respuestas disjuntas entre sí.

Al llegar a un determinado punto la búsqueda le va a indicar otra característica para la cual esa persona no tiene respuesta por lo que la búsqueda se vera truncada y el usuario no obtendrá ninguna respuesta, ni siquiera una serie de familias entre las que se puede encontrar dicha planta por lo menos para poder seguir investigando con otros libros.

Un ejemplo de búsqueda puede ser el siguiente:

35[34]. Flores sésiles, agrupadas en capítulos, rodeado por un involucre de brácteas .....	36
– Flores pediceladas –a veces con pedicelos muy cortos–, agrupadas en cimas o en umbelas ± compactas .....	37

Llegado un punto en que tu aún no has encontrado la familia de plantas a la que pertenece, el sistema dicotómico te pregunta acerca de la característica de las plantas, y en tu caso la planta a determinar no tiene flores por lo que el sistema aparte de no darte otra opción no te permite elegir otra característica en vez de contestar a la característica de las flores.

En este caso no obtenemos ninguna respuesta y además como al ser buenas todas las respuestas a las anteriores preguntas te ha conducido a esta última pregunta a la que no tienes opción de respuesta pues no puedes seguir y por tanto esta búsqueda no te proporciona ninguna respuesta, ni te permite continuar con otras preguntas, simplemente la búsqueda a finalizado sin ninguna respuesta por parte de este método.

Por ello este método de búsqueda dicotómica es ineficiente en la mayoría de los casos ya que en muchos de los casos la persona que quiera determinar alguna planta y no conozca alguna de las respuestas que este sistema te haga nunca obtendrá la respuesta deseada, es decir, una familia a la que pertenezca dicha planta que el usuario desee determinar.

Por esta razón nuestro proyecto se basa en una determinación de las plantas, en las diferentes familias que existan dando al usuario la posibilidad de no responder a determinadas características de la planta a determinar debido o al desconocimiento de la característica en cuestión o a la ausencia de esta en la misma.

Además como ayuda innovadora proporcionamos al usuario la posibilidad de conocer la descripción de dicha característica ya que pensamos que puede existir alguna persona que en un momento dado quiera determinar una planta, para conocer la familia a la que

pertenece y por el desconocimiento en la materia le resulte una tarea muy difícil. En este caso añadimos una ayuda con las distintas descripciones de las características que nosotros consideramos importantes, a la hora de discretizar en las familias, que el usuario introduzca para obtener la respuesta deseada.

## **HERRAMIENTA NITIDA**

Decidimos empezar por crear un programa que aun preguntándole una serie de características de las familias puedas no responder a ella si no conoces la respuesta. Es decir, tenemos una serie de características que hemos considerado importantes que el usuario responda para poder proporcionarle una respuesta más precisa, pero si dicho usuario no conoce la respuesta a una determinada característica no es completamente necesario introducir una respuesta en caso de no saberla.

Por ello hemos conseguido una herramienta capaz de proporcionar una respuesta aun sin conocer alguna de las respuestas. Dicha respuesta puede ser una determinada familia a la que pertenece o una serie de ellas entre las que se encuentra la planta a determinar.

Como dato a tomar en cuenta es importante conocer que hemos sido las pioneras en proporcionar una herramienta de determinación de las familias de la flora ibérica que es capaz de realizar una búsqueda de la familia a la que pertenece una determinada planta proporcionando una respuesta al usuario con la o las familias entre las que se encuentra la planta a determinar.

En ningún momento se quedara sin respuesta el usuario en el caso de no conocer alguna de la característica de la planta, ya que nuestra herramienta es capaz de proporcionar una respuesta incluso sin responder a alguna de estas preguntas acerca de las características. También como información proporcionamos un grado de pertenencia de las familias que han quedado aun como posibles a ser la familia entre las que se encuentre la planta a determinar a tener en cuenta ya que dichos grados de pertenencia están ordenados de mayor a menor, con lo que la familia que se muestra inicialmente es la que mayor posibilidad de ser la búsqueda tiene.

Nuestra herramienta ha sido la pionera en proporcionar una herramienta de determinación de plantas a los usuarios que obtenga respuesta siempre, independientemente de responder a todas las preguntas o no. Ya que dicha herramienta puede ser usada por personas ajenas a este campo de conocimiento y tendrían porque conocer el significado de todas las características preguntadas como podría tener un experto en la materia.

Con esta herramienta todas las personas teniendo o no conocimiento en el campo de la biología podría mediante una serie de preguntas obtener la familia a la que pertenece la planta la cual quiere determinar.

Además al ser una herramienta de uso público cualquier persona solo con querer realizar una determinación de una planta desconocida para él puede usarla, debido a que nuestra intención ha sido colgarla en una página de Internet de difusión mundial y con acceso publico. Incorporamos una ayuda en dicha página Web para que las personas que quieran hacer uso de nuestra herramienta sin conocer como usar un programa informático, o simplemente sin conocimientos en el campo de la informática sea capaz de determinar la planta que desee y obtener la respuesta.

# HERRAMIENTA INICIAL

La herramienta ha ido evolucionando pero una primera fase, la cual se trata de mostrar una serie de familias e ir eliminándolas según el usuario va introduciendo una serie de características, fue nuestro paso inicial.

Para ello necesitamos realizar un estudio acerca de las características de las diferentes familias de las que consta la flora ibérica.

En un primer momento nos resulto difícil ya que es un campo que no conocíamos, ya que los para nosotras resulto ser un desconocido el mundo de las plantas y mucho mas las características con las que un biólogo es capaz de determinar las plantas.

Con el documento usado por los biólogos para su determinación de plantas realizamos dicho estudio.

Inicialmente encontramos muchas dificultades porque el documento es muy extenso ya que proporciona toda la información acerca de cada familia y dentro de dicha familia posee también información sobre cada planta perteneciente a dicha familia.

Con lo que para comenzar hicimos un estudio acerca de las características de las familias. Esto tampoco resultó nada fácil ya que las familias de las consta la flora ibérica son unas 120.

Aunque nos resulto muy costoso conseguimos crear una tabla, un documento Excel, con las características de las familias así como la descripción de cada una de estas para cada familia.

Este documento consta de las siguientes características:

- **Tipo de hoja**, esta característica se divide en perenne y caduca. Estas características son muy conocidas.

- **Pelos**: las plantas poseen pelos que se pueden dividir en glandulíferos o no glandulíferos. También pueden ser estrellados

- **Tallos**: los tallos pueden ser de diferente manera. A mencionar como más importantes: erecto, postrado, trepador, ramoso.

- **Conos**: los conos pueden ser con escamas y distintos si la planta posee género masculino o femenino.



- **Hojas:** las hojas pueden ser opuestas, alternas, simples, estipulas, indusio, enteras, divididas, con soros. También poseen algunas características o descripción de estas según el género al que pertenezcan.

- **Flores:** las flores pueden ser hermafroditas, unisexuales, irregulares, solitarias y poseer una determinada inflorescencia, como por ejemplo cimosa.

Además pueden ser hipoginas, periginas, epiginas, actimorfas, solitarias e irregulares. También pueden ser bracteadas o bracteoladas.

- **Cáliz:** el cáliz consta de sépalos, y los sépalos pueden tener diferente número de soldados pueden ser lobulados, persistentes, libres, caducos, bilabiado.

- **Corola:** la corola posee lóbulos con distinto número de estos, bilabiada, etc....

- **Pétalos:** los pétalos que puede poseer la familia pueden tener diferente número. También pueden ser libres, valvados, imbricados.

- **Estambres:** los estambres en diferente número de posesión de estos, de diferentes formas: didinamos, obiplostemos, alternados, opuestos,.. También pueden estar insertados en el tubo de la corola o no.

- **Filamentos:** los filamentos pueden ser libres, cortos, nulos, soldados, ...

- **Anteras:** las anteras poseen tecas: bitecas, ditecas, .. Poseen polen en distintas formas como por ejemplo esferoidal, ...

- **Disco epiginio:** el disco epiginio que puede poseer forma aplanada o cónica, anular, cupuliforme,... y puede estar situado alrededor de la base del ovario, o confluyente en la base de los estilos,....

- **Ovario:** el ovario puede ser supero o infero y puede ser bicarpelar, tricarpelar, .... Posee loculos en distinto número: unicular, bilocular, tetralocular, También puede ser sincárpico.

- **Carpelos:** los carpelos pueden tener distinto número de ellos.

- **Estigma:** el estigma puede ser lobulado en diferente número, capitado, lineal, ...

- **Fruto:** el fruto puede estar o no encapsulado. También puede tener distinto tipo de dehiscencia: dehiscencia explosiva, indehiscente, ...

- **Semillas:** las semillas pueden poseer diferente número de estas. También poseen o no endosperma y de distinto tipo.

- **Raíces:** las plantas pueden tener diferente numero de raíces: numerosas, nulas, ...
- **Tipo de árbol:** el tipo de árbol puede ser arbusto, hierbas, leñosa, liana, árbol, acuática, trepadora,... También puede poseer rizoma y ser isosporeos o heterosporeos.
- **Estilos:** Pueden tener diferente número de estos y pueden ser cortos, libres, ...
- **Prótalos:** los prótalos pueden o no poseerlos.
- **Ramas:** las ramas pueden ser alternas, aladas, opuestas, ...
- **Primordio:** el primordio puede ser seminal apical o subapical, ...

Con tanta variedad decidimos buscar las características que consideramos no así como importantes pero si las que mayor nos pudieran discretizar las familias. Es decir, las que con pocas preguntas el usuario pudiera obtener la familia a la que pertenece la planta que quiere determinar.

Para ello realizamos un estudio a fondo de las características que lograsen hacer esto según la tabla que habíamos creado que tiene como columnas todas las familias y como filas todas las características. En la conjunción de ambas, columna con fila, la descripción que corresponda.

# CARACTERÍSTICAS SELECCIONADAS

Mediante este estudio conseguimos obtener las caracterizas con mayor índice de discretización consiguiendo obtener las siguientes:

- **Tipo de hoja:** caduca, perenne.
- **Hojas:** opuestas, alternas, con o sin estipulas, enteras, divididas, simples.
- **Flores:** hermafroditas, unisexuales, hipoginas, periginas, epiginas actimorfas, solitarias, irregulares.
- **Polen:** esferoidal, granos tricolporados.
- **Ovario:** supero, ínfero.
- **Corola:** bilabiada.
- **Fruto:** encapsulado
- **Pelos:** glanduliferos o no, estrellados.
- **Tipo de árbol:** arbusto, árbol, leñosa, liana, isosporeos, heterosporeos, herbacea, trepadora, acuática, con rizoma.

Después de obtener estas características logramos obtener una determinación de las plantas según estas características permitiéndonos discretizar lo máximo posible para obtener teniendo en cuenta las respuestas del usuario a dichas características obtener un numero mínimo de familias a las que puede pertenecer una planta a la que la estas sometiendo a una determinación para obtener la familia a la que pertenece.

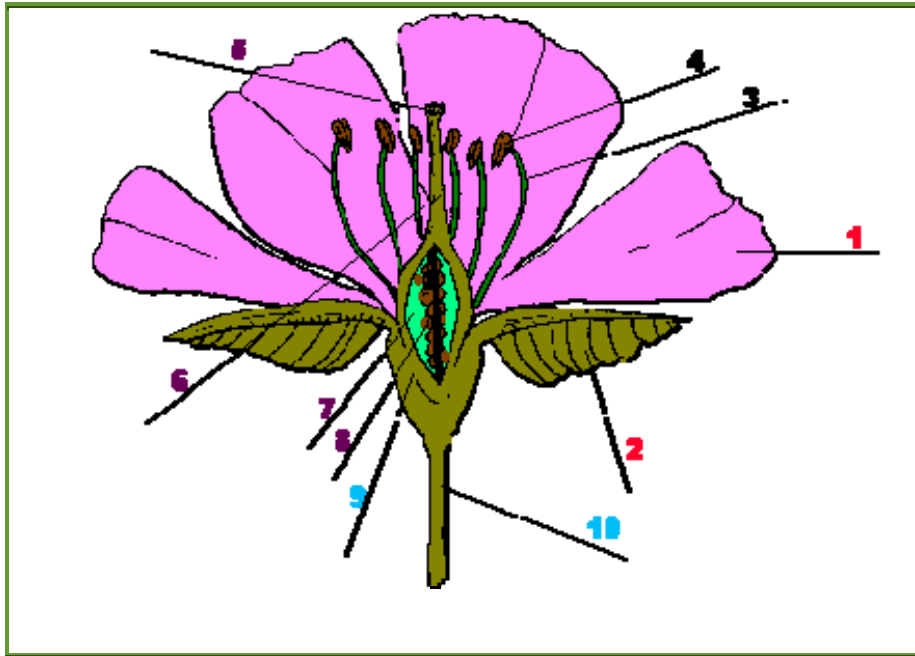
Una vez obtenidas estas características mediante este criterio debemos hacer un estudio a fondo acerca del significado de cada una de ellas.

Daremos a continuación una descripción de cada una de ellas.

# DESCRIPCIÓN DE LAS CARACTERÍSTICAS

## LA FLOR:

### PARTES



PERIANTO: La Corola. Pétalos(1).El Cáliz. Sépalos (2)

EL ANDROCEO: Estambres. Filamento (3). Antera (4)

EL GINECEO: Carpelo. Estigma (5). Estilo (6). Ovario (7). Óvulos (8)

EL EJE FLORAL: Tálamo (9). Pedúnculo (10)

El cáliz es la parte verde de la flor. Tiene una consistencia más fuerte que la corola y a sus piezas les llamamos sépalos.

La corola está formada por los pétalos que son las piezas coloreadas de las flores. Su función es atraer a los animales portadores del polen. La corola es la parte de la flor que convierte a este órgano en algo tan atractivo para los insectos y el principal motivo por el cual cultivamos las flores de jardinería.

Dentro del cáliz, y rodeado por la corola, se encuentra el androceo o parte masculina de la flor. El androceo está constituido por los estambres que unas hojas que se han

transformado con la finalidad de llevar el polen. Cada estambre consta de un filamento, que es el fragmento más alargado; y la antera que es una " especie de bolsa ", donde están encerrados los granos de polen.

Rodeado por el androceo, se encuentra el gineceo. El gineceo es la parte femenina de la flor. Está formado por uno o varios pistilos que son órganos parecidos a una botella. Cada pistilo consta de un estigma que está situado en la parte superior en forma de receptáculo para recoger el polen. El estilo que sirve de tubo conductor hacia el ovario El ovario que es la parte inferior más ampliada y donde se encuentran los óvulos que han de ser fecundados por el polen masculino.

La mayoría de las flores son hermafroditas, es decir poseen órganos masculinos y femeninos a la vez. Algunas flores solamente son masculinas y otras son femeninas. La mayoría de las plantas poseen flores hermafroditas. Hay plantas, como el roble, que poseen flores masculinas y femeninas separadas en la misma planta, y otras plantas, como el acebo, que poseen flores masculinas en una planta y flores femeninas en otra planta de la misma especie.

El pedúnculo floral une la flor a la rama.

Para que una flor se transforme en frutos debe estar previamente polinizada. La polinización es el paso del polen desde el aparato masculino de las plantas al aparato femenino. Este proceso se puede realizar a través de los animales que transportan el polen de una planta a otra o a través del viento que arrastra el polen y lo deja caer en otra planta. Más raramente se produce la autopolinización entre las flores de una misma planta o dentro de una misma flor.

**EL PERIANTO:** Constituye la parte no reproductiva de la flor. Está formada por dos tipos de piezas.

- la **corola** que está formada por los pétalos que son las piezas coloreadas de las flores. Su función es atraer a los animales portadores del polen.

-El **cáliz** que es la parte verde de la flor. Tiene una consistencia más fuerte que la corola y a sus piezas les llamamos sépalos.

A veces los pétalos y los sépalos tienen el mismo color, entonces les llamamos **tépalos**.

**EL ANDROCEO:** Es la parte masculina de la flor. Está constituida por los estambres que no son otra cosa que unas hojitas que se han transformado con la finalidad de llevar el polen. Cada estambre tiene dos partes:

- El **filamento** que lleva encima una especie de " bolsita" encima de él cargada de polen.
- La **antera** que es la "bolsita" superior donde están encerrados los granos de polen.

**EL GINECEO:** El carpelo es la parte femenina reproductora de la flor. Es una hoja que se ha modificado y que aún conserva su color verde. Los carpelos pueden estar unidos formando un solo pistilo o separados. Cada pistilo consta de las partes siguientes:

- El **estigma** que está situado en la parte superior en forma de receptáculo para recoger el polen.
- El **estilo** que sirve de tubo conductor hacia el ovario

-El **ovario** que es la parte inferior más ampliada y donde se encuentran los óvulos que han de ser fecundados por el polen masculino..

**EL EJE FLORAL:** Es la estructura que soporta las partes de la flor. Además de aguantar las piezas florales protege los óvulos de los animales. Tiene forma de copa y se llama tálamo o receptáculo. Entre éste y la ramita se encuentra el pedúnculo.

Ahora explicaremos la diferencia entre **hoja caduca** y **hoja perenne**:

Que una hoja sea perenne o caduca es el resultado de una adaptación de la planta al medio y depende fundamentalmente de las condiciones climáticas en que la planta se desarrolla y de las peculiaridades del suelo en donde se encuentra enraizada.

En general podemos decir que el hecho de mantener las hojas encima de los árboles supone un gasto de nutrientes para la planta, innecesario cuando estas no desempeñan función alguna. Por este motivo la mayoría de los árboles de los bosques de clima atlántico son de hoja caduca. No tiene sentido que un haya (*Fagus sylvatica*) o un castaño ( *Castanea sativa* ) mantengan las hojas encima del árbol cuando, por las condiciones ambientales el

árbol no puede absorber los nutrientes del suelo, dado que este normalmente se encuentra helado. Estos árboles entran en una fase de reposo hiberna l y dejan caer las hojas.

También existen hojas **semicaducas**:

Plantas con hojas semicaducas, que reducen la amplitud de estos órganos, quedando en estado semisequedad encima de la planta o dejando caer parte de ellas en condiciones ambientales secas. Cuando las condiciones son normales, recuperan rápidamente su volumen y cantidad. Las estepas (Cistus ) presentan estas características.


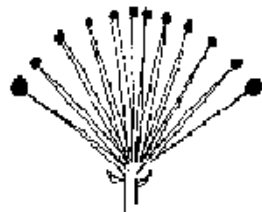


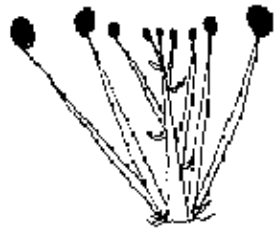
Ahora explicaremos las **inflorescencias**:

Las flores no se presentan en la mayoría de los casos aisladas. Normalmente aparecen agrupadas de una determinada manera que denominamos inflorescencias. Las inflorescencias pueden ser:

Simples: cuando presentan la misma estructura todas las flores

Compuestas: cuando dentro de una estructura compleja tenemos otras estructuras simples.

### PRINCIPALES INFLORESCENCIAS SIMPLES

Racimo	Umbela	Espiga	Capítulo	Corimbo
				

En los racimos las flores están colocadas a lo largo de un eje floral con los pedúnculos parecidos, las flores más jóvenes en la parte superior y las más viejas en la inferior.


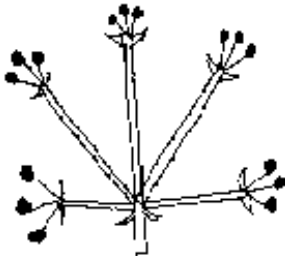
Las **umbelas** presentan flores que parecen salir del mismo punto y llegar a la misma altura.

Las **espigas** son inflorescencias parecidas al racimo pero con las flores sin pecíolo.

En los **capítulos** las flores se agrupan encima del receptáculo floral que se parece a una plataforma

En los **corimbos** todas las flores están colocadas a lo largo de un eje floral pero, a diferencia del racimo, los pedúnculos tienen diferente longitud de forma que todas las flores quedan a la misma altura.

### PRINCIPALES INFLORESCENCIAS COMPUESTAS

Panícula	Umbela compuesta
	

Las **panículas** están formadas por racimos agrupados.

Las **umbelas compuestas** son inflorescencias que tienen agrupadas algunas umbelas.

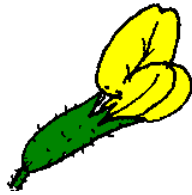


## Corola:

- Según el cáliz:



Flores tubulosas cuando el cáliz tiene forma de tubo...



Flores bilabiadas cuando el cáliz presenta dos labios .Uno más grande que el otro.



Flores vesiculosas cuando el cáliz está hinchado.



Flores dialisépalas si tienen el cáliz separado con toda claridad.



Flores gamosépalas si el cáliz está unido total o parcialmente.

- Según la posición del ovario:



**Ovario súpero:** con el ovario situado por encima del resto de los elementos florales.



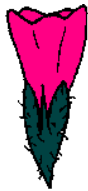
**Ovario seminífero:** con todos los elementos alrededor del ovario.



**Ovario ínfero:** con el ovario situado por debajo del resto de los elementos.

## LA FLOR:

- Según la corola:



**Gamopétalas:** Si los pétalos están soldados total o parcialmente.



**Dialipétalas:** Si los pétalos no están unidos.



**Actinomorfas:** Se pueden dividir en dos partes exactas (simetría bilateral)



**Cigomorfas o zigomorfas:** Un solo plano de simetría. Un lado siempre es diferente del otro (sin simetría bilateral)

#### Algunos tipos de gamopétalas:

infundibuliforme	bilabiada	tubulosa	acampanada





Flores **infundibuliformes** son aquellas que tienen la corola en forma de embudo.

Flores **bilabiadas** son las que tienen la corola partida en forma de dos labios.

Flores **tubulosas** son las que tienen la corola en forma de tubo.

Flores **acampanadas** son aquellas que tienen la corola en forma de campana.

#### Algunos tipos de dialipétalas:

papaveráceas	papilionáceas	crucíferas	Rosáceas
			

Flor **papaverácea** es la que tiene cuatro pétalos distribuidos en dos verticilos, como la amapola.

Flor **papilionácea** es aquella que por su forma nos recuerda a una mariposa.

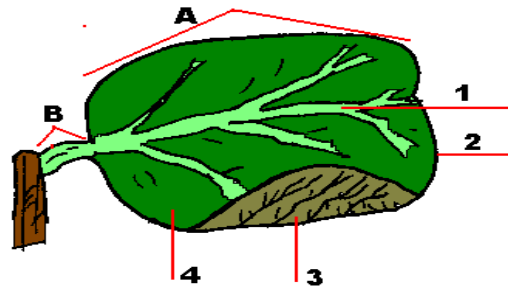
Flor **crucífera** es la que tiene cuatro pétalos en forma de cruz.

La flor **rosácea** tiene cinco pétalos, a la manera de la rosa silvestre.

La hoja es una de las partes más importantes de los vegetales puesto que es la parte de la planta que está encargada de realizar la función clorofílica , así como la respiración y la transpiración vegetal.

## LA HOJA:

Hay muchos tipos de hojas que permiten distinguir unas plantas de otras , pero, esencialmente, toda hoja esta formada por las partes siguientes:

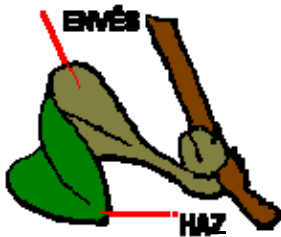


### A) LIMBO

1.-NERVIOS 2.-CONTORNO 3.-ENVÉS 4.-HAZ

El limbo es la parte ancha de la hoja. Es su parte más vistosa y lo que la mayoría de la gente entiende e identifica como hoja cuando se menciona tal nombre.

### B) PECÍOLO

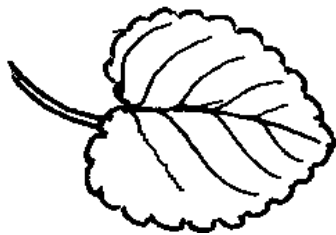


Dentro del limbo hemos de hablar de:

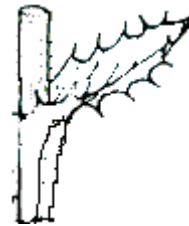
- El **haz**: Es la parte superior de la hoja. Suele tener un color verde brillante.
- El **envés**: Es la parte opuesta al haz. Su color es normalmente más oscuro y presenta muchas veces pelos.
- Los **nervios**: son una especie de arrugas o canales que recorren el limbo de la hoja. En realidad, son los vasos conductores que discurren a lo largo de su superficie.

-El **contorno**: Constituye el margen o extremo del limbo. Puede ser de diferentes formas que se utilizan para distinguir unas hojas de otras.

El **pecíolo** es la parte de la hoja que une el limbo a la rama. Tiene forma de rabito y, a través de él, discurren los vasos conductores. Hay algunas hojas que no tienen pecíolo. Estas hojas sin pecíolo se llaman sésiles.



**Pecioladas**

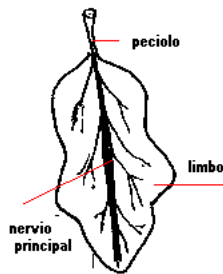


**Sésiles**

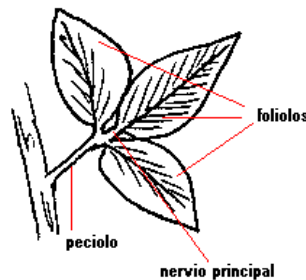
Son **hojas pecioladas** las que tienen pecíolo que puede tener diferentes tamaños, considerados normales, largos o cortos

Son **hojas sésiles** aquellas en las que el limbo sale directamente de la ramita. No tienen pecíolo.

- **Según el limbo:**



**Hoja simple**



**Hoja compuesta**

Son hojas simples las que tienen un limbo sin partir o, aunque este limbo esté partido, las divisiones no llegan hasta el nervio principal.

Son hojas compuestas aquellas en las que el limbo está dividido en fragmentos que llegan al nervio principal. A cada una de estas partes, que son como hojas, les llamamos folíolos.

- **Según el margen:**



Son **hojas enteras** las que tienen el margen liso.

Son **hojas onduladas** las que tienen pequeños entrantes con márgenes suaves como olas.

Son **hojas dentadas** las que tienen pequeños dientes al margen.

Son **hojas aserradas** las que tienen pequeños dientes inclinados como una sierra.

Son **hojas lobuladas** las que presentan entrantes y salientes redondeados.

- **Según la forma del limbo:**



**Hoja elíptica** es la que tiene forma de elipsis.

**Hoja lanceolada** es la que tiene forma de lanza.

**Hoja acicular** es la que tiene forma de aguja.

**Hoja oval** es la que tiene forma de huevo.

**Hoja acorazonada** es la que tiene forma de corazón.

**Hoja sagitada** es la que tienen forma de alabarda, que era un arma antigua que combinaba la lanza con el hacha.

**Hoja lineal** es la que es estrecha y alargada como una cinta.

- **Según la nervadura:**



**Hojas paralelinervias** son aquellas en las que los nervios son aproximadamente paralelos unos de otros.

**Hojas peninervias** son aquellas en las que aparece un nervio principal del que salen los nervios secundarios como si se tratase de las barbas de una pluma.

**Hoja palminervias** son aquellas en las que los nervios de la hoja nos recuerdan la forma de la palma de la mano. Del centro de la palma salen los nervios hacia fuera como si fuesen los dedos



- Según su disposición en el tallo:



**Hojas alternas** son aquellas que salen en una cantidad de una por cada nudo y cada vez en un lado opuesto del tallo.

**Hojas opuestas** son las que están colocadas en un número de dos por nudo y cada una en el lado opuesto del tallo de la otra.

**Hojas connatas** son las que están colocadas como las opuestas, pero no tienen pecíolo.

**Hojas verticiladas** son las que nacen a la misma altura del tallo en grupos de más de tres y van describiendo como una especie de anillo alrededor del tallo.

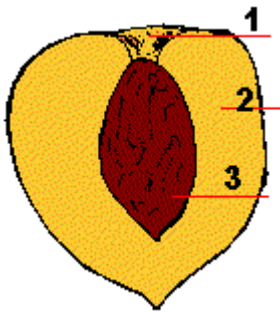
**Hojas basales** son las que salen de la parte baja de la planta a ras del suelo.

## LOS FRUTOS

El fruto es la parte de los vegetales que está a cargo de proteger las semillas y asegurar su dispersión. Es el resultado de la fecundación del ovario, especialmente por el engrosamiento de las paredes de éste, aunque algunos frutos tienen otro origen ya que pueden proceder del engrosamiento del receptáculo floral o de otro lugar de la flor.

Muchos son los tipos de frutos que tenemos, pero vamos a ver cuales son las partes de un fruto muy típico, el carnosos:

### PARTES DE LOS FRUTOS



A) **Pericarpo** 1 exocarpo o epicarpo 2 mesocarpo 3 endocarpo

El **pericarpo** es la cubierta de la semilla. Es todo aquello que rodea a la semilla. Consta de tres partes:

- El **exocarpo** o epicarpo que es la parte más externa del fruto. En una manzana, por ejemplo, sería lo que conocemos como piel.

- El **mesocarpo** es la parte más gruesa de la mayoría de los frutos. En un melocotón, por ejemplo, sería "la carne" que nos comemos.

- El **endocarpo** es la parte normalmente endurecida que cubre la semilla. En una ciruela, por ejemplo, sería "el hueso".

## **A) Semilla**

La semilla se encuentra encerrada dentro del endocarpo. La estudiaremos en detalle en otra sección.

## **TIPOS DE FRUTOS: FRUTOS CARNOSOS**

Son aquellos que tienen el pericarpo más o menos blando. Dicho de otra manera aquellos que tienen suficiente "carne" alrededor del hueso. Como ejemplos de frutos carnosos tendríamos los siguientes:

### **Bayas**

Son frutos con el epicarpo muy blando y con el mesocarpo y endocarpo muy carnosos.

### **Hesperidios**

Son frutos que presentan materia carnosa entre el endocarpo y las semillas. Los carpelos se encuentran cerrados.

### **Pepónides**

Son frutos procedentes de un ovario ínfero con la parte exterior del pericarpo endurecido.

### **Drupas**

Son frutos con el mesocarpo carnoso y el endocarpo endurecido en forma de hueso, dentro del cual se encuentra la semilla.

### **Polidrupes**

Son frutos que combinan diversas drupas a la vez.






### **Pomos**

Son frutos de mesocarpo carnoso y endocarpo coriáceo. Se originan de un ovario ínfero.

## TIPOS DE FRUTOS: FRUTOS SECOS

Son frutos secos aquellos que no tienen una textura blanda cuando están maduros. Tienen una apariencia como de madera y al presionar sobre ellos no están blandos. Determinamos los frutos secos de dos maneras:

### Indehiscentes

Aquenio	Núcula	Cariópside	Sámara	Nuez
				

Los **aquenios** son frutos de una sola semilla, no soldada dentro del carpelo. A veces los aquenios aparecen unidos formando lo que se conoce como poliaquenios (diaquenio, biaquenio etc.)

Las **núculas** son frutos con el pericarpo endurecido y recubierto por una cúpula en la base.


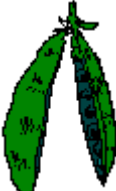



Las **cariopsides** son frutos en los que la semilla esta pegada al pericarpo.

Las **sámares** son aquenios provistos de alas.

Las **nueces** son frutos de una sola semilla con protección exterior dura.

### Dehiscentes

Son aquellos frutos que se abren al madurar y dejan escapar las semillas.

Folículo	Legumbre	Silicua	Cápsula	Pixidio
				

Los **folículos** son frutos que están formados de un solo carpelo. Cuando maduran se abren por un lado.

Las **legumbres** también son frutos derivados de ovarios de un solo carpelo. A diferencia de los folículos se abren por dos lados.

Las **silículas** provienen de dos carpelos unidos que, al madurar, se abren.

Las **cápsulas** son frutos procedentes de ovarios compuestos con mucha variedad de aperturas.

Los **pixidios** son cápsulas que se abren transversalmente.

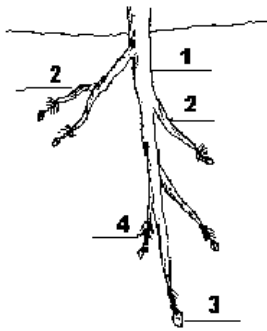
## LA RAÍZ

La raíz es una parte muy importante de la planta que tiene dos funciones principales:

Sujetar la planta al suelo

Succionar el agua y las sales minerales del suelo

Aquí tienes el dibujo de una raíz típica



### 1 ) Raíz principal

La raíz principal es la raíz más gruesa.

### 2) Raíces secundarias

Las raíces secundarias salen de la raíz principal y no son tan gruesas como aquella.

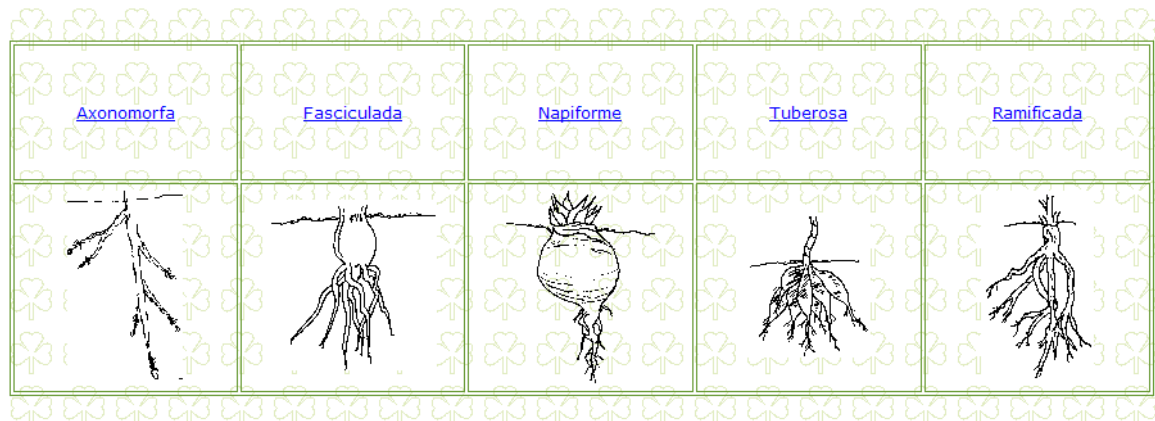
### 3) Caliptra

La caliptra es la protección con la que terminan las raíces. Sirve para que las raíces puedan perforar el suelo. Actúa, por lo tanto, como una especie de taladro natural.

### 4) Pelos absorbentes

Los pelos absorbentes son unos filamentos diminutos que recubren las raíces y tienen la función de absorber el agua y las sales minerales del suelo.

## TIPOS DE RAÍCES



**Raíces axonomorfas** son las que están formadas por una raíz principal más gruesa y otras que salen de la principal más delgadas. Es la raíz típica que siempre se utiliza como modelo cuando se estudia esta parte de la planta.

**Raíces fasciculadas** son las raíces que no tienen una raíz principal. Todas presentan, más o menos, el mismo grosor.

**Raíces napiformes** son aquellas en las que la raíz principal es muy gruesa porque acumula sustancias de reserva.

**Raíces tuberosas** son raíces de estructura fasciculada en las que se producen engrosamientos por acumulación de sustancias de reserva.

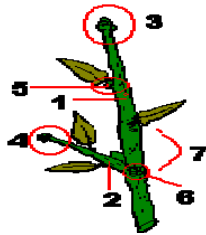
**Raíces ramificadas** son aquellas que, sin tener una raíz principal, nos recuerdan por su forma las ramas de un árbol.

## EL TALLO

Es una parte muy importante de la planta que tiene una doble finalidad:

- 1) Sostener las flores y las hojas a una altura determinada del suelo.
- 2) Transportar la savia desde las raíces a las partes superiores de la planta.

## PARTES DEL TALLO



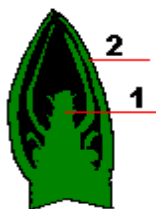
### 1) Tallo principal

El tallo principal es el tallo más importante de la planta. De él comienzan a salir los tallos secundarios. Este tipo de tallo es muy evidente en las hierbas y en ciertos tipos de árboles como los abetos y los pinos.

### 2) Tallo secundario

Los tallos secundarios salen a medida que se va desarrollando el tallo principal.

#### 1) Yema principal





Las yemas tienen la función de realizar el crecimiento de los tallos. Están formadas con el cono vegetativo **(1)** y por una serie de hojitas que lo protegen y que, más tarde, darán paso a las auténticas hojas. Estas hojitas se llaman primordios foliares **(2)**. Las yemas son partes muy delicadas y sensibles al frío. Por eso en invierno están protegidas por una cubierta de hojas endurecidas.

La yema principal es la que tiene la función del crecimiento vertical de la planta. Esta formada por el ápice vegetativo protegido por el conjunto de hojitas llamadas primordios foliares.

#### **4) Yema secundaria**

Las yemas secundarias salen al extremo de los tallos secundarios. Son las encargadas del crecimiento de los tallos secundarios.

#### **5) Yema axilar**

Las yemas auxiliares están situadas en los nudos de los tallos. De ellas salen las hojas y las flores.

#### **6) Nudo**

Los nudos son unos engrosamientos situados en los tallos. A su altura es donde nacen las hojas.

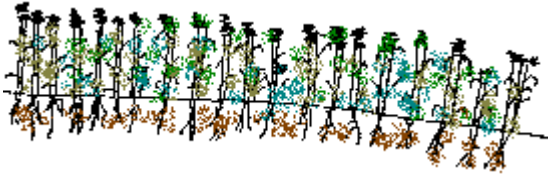
#### **7) Entrenudo**

Los entrenudos son los espacios situados entre los nudos

## TIPOS DE TALLOS

- **TALLOS HERBÁCEOS:**

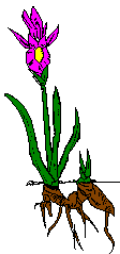
Son aquellos tallos que no han desarrollado estructuras leñosas endurecidas. Su consistencia es blanda. Normalmente son hierbas:



**Anuales:** Son aquellas plantas en las que el ciclo vegetativo dura un año. Después se secan. Así lo hacen las que conocemos como "hierbas".



**Vivaces:** Son aquellas plantas cuyos tallos también duran un solo año, pero mantienen partes subterráneas de las que vuelven a brotar nuevos tallos al año siguiente.



- **TALLOS LEÑOSOS:**

Son aquellos que han desarrollado estructuras endurecidas. Aquello que normalmente conocemos como "madera". Según su altura los determinamos en:

**Árboles:** son aquellas plantas de tallo leñoso con una altura superior a cinco metros . En este caso los tallos se conocen con el nombre de troncos, los cuales no se ramifican hasta una altura considerable del suelo.



**Arbustos:** son aquellas plantas de tallo leñoso que miden entre uno y cinco metros de altura. La ramificación en este caso comienza a nivel de tierra.



**Matas:** son aquellas plantas de tallo leñoso con una altura inferior al metro.



**Hierbas:** son aquellos tallos que no han desarrollado estructuras leñosas endurecidas. Su consistencia es blanda.

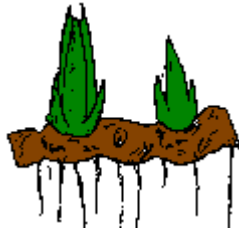


### • OTROS TIPOS DE TALLOS:

Habría todo un conjunto de plantas que no se adaptarían a las determinaciones más generales vistas hasta ahora. Así tendríamos:

## TALLOS SUBTERRÁNEOS:

**Rizomas**



**Tubérculos**



**Bulbos**

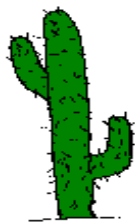


Los **rizomas** son unos tallos de crecimiento horizontal por debajo de la superficie terrestre. De estos tallos salen las raíces hacia dentro de la tierra y unos tallos herbáceos hacia afuera. Las cañas serían un ejemplo.

Los **tubérculos** son tallos engrosados por acumulación de sustancias nutritivas, como la patata, por ejemplo.

Los **bulbos** son en realidad yemas adaptadas a vivir debajo de la tierra, que acumulan reservas. La cebolla sería un ejemplo.

## TALLOS SUCULENTOS



Estos tallos se engordan mucho al acumular agua. Son así una reserva de este líquido necesaria para los largos periodos de sequía. Su adaptación al medio ha llegado a transformar las hojas de la mayoría de estas plantas en espinas; lo que, además de favorecer

la retención de agua, constituye una buena defensa contra los herbívoros . Dentro de este grupo las plantas más significativas serían las cactáceas.

### **TALLOS RASTREROS**

Muchas plantas presentan tallos débiles que se arrastran por tierra y al mismo tiempo desarrollan nuevas raíces con las que producirán nuevas plantas. Este tipo de tallos se llaman tallos rastreros. Dentro de este grupo tendríamos plantas como la fresa.



### **TALLOS TREPADORES**

Las plantas que presentan tallos trepadores pueden ser:

**Volubles:** Presentan tallos flexibles que se enroscan alrededor de otras plantas. La planta de la judía sería un ejemplo.



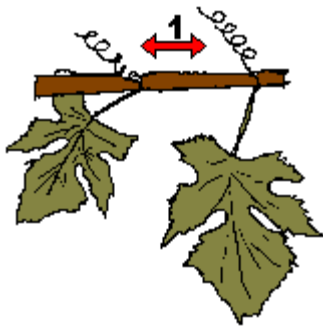
**Con raíces adherentes:** Presentan tallos que han desarrollado raíces aéreas para sujetar la planta a otras superficies. Por ejemplo la hiedra.



**Con espinas:** Son tallos provistos de espinas con la doble finalidad de sujetar la planta y defenderla contra el ataque de los herbívoros . Por ejemplo la zarza.



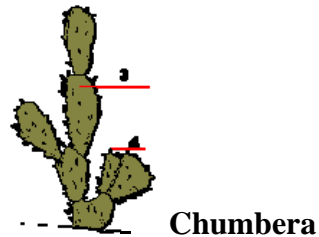
**Con zarcillos (1) :** Son tallos no principales transformados para enrollarse alrededor de un objeto. Por ejemplo la vid.



## OTROS TALLOS

### Rusco

Los **filocladios** y los **cladodios** son tallos con el aspecto de hojas. Aparecen porque las hojas son muy pequeñas o porque se han transformado y ya no pueden cumplir con su función. Dos ejemplos muy claros serían el rusco y la chumbera.



1) **filocladios**

2) **hojas auténticas diminutas.**

3) **cladodios.**

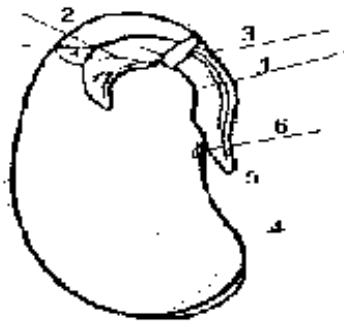
4) **hojas transformadas en pinchos.**

## LA SEMILLA

Las semillas son óvulos maduros de los cuales, de darse las condiciones oportunas, nacerán nuevas plantas.

### PARTES DE LA SEMILLA

Esta es una típica semilla. Podemos ver las partes siguientes:



#### A) embrión

El **embrión** es la pequeña planta en estado embrionario. Cuando las condiciones son favorables (adecuada humedad, calor y oxígeno) se desarrolla dando lugar a una nueva planta. Contiene las partes siguientes:

##### 1) radícula

La radícula es la parte del embrión que emerge primero. Una vez fuera se convierte en una auténtica raíz, produciendo pelos absorbentes y raíces secundarias.

##### 2) plúmula

La plúmula es una yema, se encuentra a lado opuesto de la radícula



### **3) hipocotilo**

### **4) cotiledón**

Cotiledones, que adquieren la función de primeras hojas o de reserva alimenticia, a veces ambas cosas a la vez.

De acuerdo al número de cotiledones, determinamos las plantas en: monocotiledóneas (con un solo cotiledón) o dicotiledóneas (con dos cotiledones). En el primer grupo encontramos plantas tan importantes como los cereales, palmeras, lirios, tulipanes u orquídeas. Los miembros del segundo grupo son más numerosos y comprenden la mayoría de las angiospermas.

### **B) endospermo**

**El endospermo o albumen** es la reserva alimentaria contenida en la semilla. En las monocotiledóneas esta constituido por almidón, conformando casi la totalidad de la semilla. A veces esta reserva se encuentra incluida en los cotiledones, como ocurre siempre en el caso de las dicotiledóneas.

### **C) epispermo (5)**

Epispermo es la cubierta exterior. Esta formada por la testa y, en el caso de las angiospermas, con una cubierta suplementaria por debajo de esta, llamada tegmen. La testa a veces es delgada, como ocurre en las semillas protegidas por el endocarpio leñoso, pero a veces, cuando falta esta protección, la testa actúa de defensa contra el mundo exterior además de evitar la pérdida de agua de la semilla. Sobre esta superficie, podemos ver el micrópilo que es como un pequeño poro, a través del cual se había producido la entrada del tubo polínico en el óvulo y por donde se dirige la radícula en la germinación.

### **6) Micrópilo**

## TIPOS DE PLANTAS SEGÚN SU CONSTITUCIÓN

En la determinación de las plantas se utilizan una serie de criterios para hacerla posible. Los criterios básicos que se utilizan son:

- Si poseen o no poseen raíces, tallo y hojas.
- Si poseen o no poseen vasos conductores.
- Si poseen o no poseen semillas.
- Si poseen o no poseen frutos.
- Si poseen uno o dos cotiledones.

Según estos criterios, podemos determinar las plantas en:

- **Plantas no vasculares (Briofitas)** : son plantas de estructura muy simple, llamada talo, en la que no se distingue la raíz, el tallo y las hojas. . Son plantas que suelen vivir en lugares húmedos ya que absorben el agua directamente del aire o del sustrato. Se reproducen a través de esporas y necesitan el agua para poder desarrollarse. Estas plantas representan el paso de los vegetales desde la vida acuática a la terrestre. Constituyen unas 23.000 especies. Dentro de este grupo tenemos:

- **Los musgos** (*Bryopsida*)
- **Las hepáticas** (*Marchantiopsida*)
- **Los antoceros.** (*Anthocerotopsida*)

- **Plantas vasculares (Cormofitas)** : Son aquellas plantas que poseen raíz, tallo y hojas. Presentan un sistema vascular para la distribución del agua y los nutrientes. Esta formado por el xilema o sistema vascular que distribuye la savia bruta hacia las hojas y el

floema o sistema vascular que se encarga de la distribución de la savia elaborada hacia el resto de la planta. Dentro de este grupo tenemos:

- **Los pteridofitos:** (Plantas sin flores o semillas ) Corresponde a lo que se llamaban anteriormente criptógamas. Comprenden unas 12.000 especies. Estos se determinan en:

- Helechos ( *Pteridophyta*)
- Licopodios ( *Lycopodiophyta*)
- Equisetos ( *Equisetophyta*)
- Psilofitos( *Rhyniophyta*)

- **Los espermatofitos** (Plantas con semillas) Corresponde a lo que se llamaban anteriormente fanerógamas. Los espermatofitos a su vez se determinan en:

- **Gimnospermas:** Son aquellas plantas cuyas semillas en su madurez no se encuentran encerradas en los frutos. Existen unas 850 especies. Poseen flores poco vistosas cuya polinización es realizada por el viento. Dentro de este grupo tenemos:

- Las coníferas y taxáceas (*Coniferopsida*)
- Las cícadas ( *Cycadopsida*)
- El ginkgo ( *Ginkgopsida*)
- Gnetopsida

- **Angiospermas:** Plantas cuyas semillas se encuentran encerradas en su madurez dentro de los frutos. Poseen flores muy vistosas. Se conocen unas 224.000 especies. Estas pueden ser :

- **Monocotiledóneas** ( *Liliopsida*) : Con un solo cotiledón en los embriones. Existen unas 50.000 especies.

- **Dicotiledóneas** ( *Magnoliopsida* ):

Con dos cotiledones . Existen unas 170.000 especies.

Para los amantes de las plantas y la jardinería resulta muy importante conocer cómo crecen, viven y se desarrollan nuestras plantas. En primer lugar, para principiantes, nos permitirá saber cuánto duran nuestras plantas; así si se nos muere alguna, a lo mejor no es a causa de una enfermedad o plaga sino simplemente que ha concluido su ciclo de vida o que en realidad no está muerta del todo sino "esperando tiempos mejores" para volver a desarrollarse. En segundo lugar, podríamos describir nuestras plantas o interpretar fichas botánicas que a veces nos hacen un lío cuando comienzan "... es una planta herbácea, bianual, vivaz..." o "...es un árbol o arbusto perenne siempreverde..." Y, en tercer lugar, podríamos diseñar mejor nuestro jardín teniendo en cuenta los ciclos de vida y hábitos de crecimiento de nuestras plantas.

## TÉRMINOS AÑADIDOS POR EL BIÓLOGO

Las determinaciones de las plantas más utilizadas en jardinería tienen que ver, la mayoría de las veces, con el tipo, durabilidad y adaptaciones del tallo. Quizás, una de las determinaciones más globales que existe de las plantas es dividir las, de acuerdo a su consistencia, en herbáceas y leñosas; categoría que está ligada directamente al desarrollo del tallo:

- Herbáceas: Son aquellas plantas cuyos tallos, independientemente de su tamaño, no han desarrollado estructuras leñosas por lo que su consistencia es más o menos blanda, tierna, flexible y jugosa. A la mayoría se conoce como hierbas (aunque estrictamente el término hierba se refiere a las herbáceas que mueren después de su estación de crecimiento). Ejemplos de este tipo de plantas son: col ornamental (*Brassica oleracea* L.), caléndula (*Calendula officinalis* L.), pensamiento (*Viola tricolor* L.), singonio (*Syngonium podophyllum* S.), cortaderia (*Cortaderia selloana* S.), clavel de poeta (*Dianthus barbatus*), eléboro (*Helleborus foetidus* L.), tártago (*Euphorbia characias* L.), cáñamo (*Cannabis sativa* L.), oreja de conejo (*Stachys lanata* L.), hosta (*Hosta plantaginea*), Vinca (*Vinca major* y *Vinca minor*), ciclámen (*Cyclamen persicum* L.), etc.

- Leñosas: Son aquellas plantas cuyos tallos, independientemente de su tamaño, han desarrollado estructuras leñosas (madera) por lo que su consistencia es dura y rígida debido a la presencia de un tipo de tejido llamado esclerénquima (tejido de paredes muy gruesas). A la mayoría se le conoce como árboles o arbustos y a otros como matas. Ejemplos de este tipo de plantas son: abedul (*Betula pendula* R.), sauce llorón (*Salix babylonica* L.), camaedorea (*Chamaedorea elegans* W.), granado (*Punica granatum* L.), buganvilla (*Bougainvillea glabra*), acebo (*Ilex aquifolium* L.), abeto (*Abies alba* M.), rosa de la China (*Hibiscus rosa-sinensis* L.), sauzgatillo (*Vitex agnus-castus* L.), mahonia (*Mahonia aquifolium* P.), tomillo (*Thymus vulgaris* L.), abelia (*Abelia x grandiflora* A.), ciruelo de

Japón (*Prunus cerasifera* E.), camelia (*Camelia japónica*), cycas (*Cyca revoluta* B), limonero (*Citrus limonium* L.), lavanda (*Lavandula vera*), etc.

Entre estos dos tipos generales aparecen una estructura intermedia:

- Semileñosas: Son aquellas plantas cuyos tallos han desarrollado una estructura de consistencia intermedia entre herbácea y leñosa debido a la presencia de un tipo de tejido llamado colénquima (tejido de paredes con engrosamientos irregulares). Ejemplos de este tipo de plantas son: palmera washingtonia (*Washingtonia robusta* W. y *Washingtonia filipera* W), girasol (*Helianthus annus* L.), hortensia (*Hydrangea macrophylla* S.), retama, geranio, etc.

- Suculentos: Son aquellas plantas con tallos verdes (realizan la fotosíntesis), esponjosos, poco consistentes y que están especializados en almacenar agua. El ejemplo más ilustrativo de plantas suculentas son los cactus.

De acuerdo a su ciclo de vida las plantas herbáceas pueden ser...

Por ciclo de vida entendemos el tiempo que media entre la plantación de la semilla hasta la muerte de la planta adulta, es decir, el ciclo de semilla a flor y de flor nuevamente a semilla.

- Plantas anuales: En jardinería se conocen como plantas de temporada y en el argot popular como hierbas ya que sólo duran unos cuantos meses completando su ciclo de vida en un año (única estación de crecimiento) desde que se planta la semilla hasta que muere la planta. Es decir, en un mismo año las semillas germinan, se forman todas las partes vegetativas de la planta (raíz, tallo, hojas, etc.), se produce la floración y formación de nuevas semillas y la planta muere. En este caso las semillas son altamente resistentes al frío y la desecación siendo la única parte de la planta que sobrevive. Ejemplos de plantas anuales son: girasol (*Helianthus annus* L.), caléndula (*Calendula officinalis* L.), cáñamo (*Cannabis sativa* L.), petunias, alelí, etc. Las plantas anuales se pueden subdividir en resistentes (generalmente se siembran en primavera en el lugar definitivo que van a ocupar porque resisten las condiciones adversas del clima), semi-resistentes (generalmente no

resisten las heladas por lo que se siembran previamente en un emplazamiento protegido y después se llevan al lugar definitivo sin riesgo de fríos intensos) y susceptibles (generalmente no soportan ni siquiera los fríos ligeros por lo que deben estar en interiores o invernaderos).

- Plantas bianuales: También se les conoce como bianuales o bienales. En este caso el nombre no quiere decir que la planta dure dos años sino que su ciclo de vida transcurre en dos años diferentes, o sea, una parte de su vida la desarrollan en un año y la otra parte en otro año (dos estaciones de crecimiento). Tras la germinación, y durante el primer año, se forman un tallo corto, una roseta de hojas próximas al suelo y una raíz modificada para almacenar alimento (ej.: zanahoria). Al llegar el frío el desarrollo se detiene (fase de latencia) y, durante el segundo año, al pasar el frío moviliza las reservas almacenadas, florece, da frutos, forma nuevas semillas y después muere. Ejemplo de plantas bianuales son: pensamiento (*Viola tricolor L.*), clavel de poeta (*Dianthus barbatus*), col ornamental (*Brassica oleracea L.*), etc.

- Plantas perennes: Aquellas plantas cuyas estructuras vegetativas (raíz, tallo, hojas, etc.) persisten desde la germinación año tras año (siempre más de dos) sin marchitarse ni perder sus estructuras (tallos, hojas) en ninguna época del año, o sea, tienen un ciclo de vida abierto e ilimitado. Ejemplo de plantas perennes son: hierba de la pampa (*Cortaderia selloana S.*), dieffenbaquia (*Dieffenbachia spp.*), eléboro (*Helleborus foetidus L.*), tártago (*Euphorbia characias L.*), plátano (*Musa spp.*), etc. Las plantas perennes se pueden subdividir en resistentes (puede vivir durante muchos años en el jardín soportando cualquier tipo de clima), semi-resistentes (hay que protegerlas de las duras condiciones invernales), susceptible (no soportan el frío y hay que mantenerlas en interiores o invernaderos).

- Plantas vivaces: Al igual que las perennes son plantas que pueden vivir más de dos años. Sin embargo, la vivaz se marchita con la llegada del invierno, es decir, se secan sus tallos y hojas (partes epigeas) pero sus raíces siguen vivas bajo la tierra y, cuando llegan las condiciones óptimas (ej.: primavera) vuelven a brotar. Ejemplo de plantas vivaces son: hosta (*Hosta plantaginea*), tártago (*Euphorbia characias L.*), tulipán (*Tulipa spp.*),

aglaonema (*Aglaonema spp.*), oreja de conejo (*Stachys lanata L.*), narciso (*Narcissus spp.*), etc.

¿En qué nos podemos confundir...?

En la determinación que hemos analizado hasta aquí debemos destacar los puntos en que más nos confundimos por simples que parezcan:

1. Las plantas herbáceas pueden ser anuales, bianuales, perennes o vivaces.
2. Las plantas leñosas sólo pueden ser perennes.
3. Las plantas anuales y vivaces se parecen en que las partes aéreas (epigeas) como hojas y tallos mueren al cabo de un año.
4. Las plantas anuales y vivaces se diferencian en que: cuando transcurre un año las anuales mueren completamente (tanto las partes aéreas como subterráneas) mientras que las vivaces mantienen vivas sus partes subterráneas (raíces o tallos modificados).
5. Las plantas perennes y vivaces se parecen en que siempre duran más de dos años.
6. Las plantas perennes y vivaces se diferencian en que: las perennes no se marchitan (no pierden sus tallos ni sus hojas en invierno) y las vivaces sí se marchitan durante el invierno (pierden las partes aéreas sobreviviendo la raíz).
7. Una misma planta, en dependencia del clima, puede ser anual, bianual o perenne. Por ejemplo: el pensamiento (*Viola tricolor L.*) es una planta bianual en sitios donde ocurran heladas y convertirse en bianual e incluso vivaz en sitios sin heladas. La col ornamental (*Brassica oleracea L.*) puede ser anual, bianual e incluso perenne dependiendo del clima. La capuchina (*Tropaeolum majus L.*) es una planta perenne en su hábitat natural (Perú) y anual cuando se cultiva fuera de los sitios de procedencia.

De acuerdo a las modificaciones del tallo las plantas herbáceas pueden determinarse en...



Las plantas bianuales, perennes y vivaces tienen que pasar las estaciones desfavorables (ej.: el invierno) por lo que son capaces desarrollar adaptaciones en sus tallos que le permitan, sobre todo, almacenar sustancias de reserva elaboradas en los períodos favorables en órganos como:

- Rizomas: Son tallos subterráneos (se desarrollan debajo de la superficie del suelo), de crecimiento horizontal y que pueden ramificarse. Generalmente están muy engrosados (aumentados de tamaño) para el almacenamiento además de funcionar en la reproducción vegetativa de la planta. Las ramas engrosadas tienen entrenudos, raíces adventicias y yemas propias de las que brotan (hacia el exterior) los tallos herbáceos de la planta. Crecen indefinidamente con el transcurso de los años. A las plantas que poseen esta estructura se les llama rizomatosas. Ejemplo de plantas rizomatosas son: oxalis (*Oxalis articulata* L.), helecho cuerno de alce (*Platucerium bifurcatum*), nenúfar (*Nymphaea alba* L.), etc.

- Bulbos: Son yemas modificadas adaptadas a vivir bajo la tierra, encargadas del almacenamiento y que están compuestas por hojas engrosadas y carnosas que funcionan como escamas foliares llamadas catáfilos (función reservante). Presentan una yema terminal que producirá un tallo central (foliar o florífero) del cual saldrán raíces llamadas discos (porque crecen más en grosor que en longitud) y una yema axilar que es la encargada de originar un nuevo bulbo. A las plantas que poseen esta estructura se les llama bulbosas. Ejemplo de plantas bulbosas son: tulipán (*Tulipa spp.*), cebollino (*Allium schoenoprasum* L.), narciso (*Narcissus spp.*), cebolla (*Allium cepa* L.), ajo (*Allium sativum* L.), etc.

- Tubérculo caulinar: Son tallos engrosados por la acumulación de sustancias nutritivas, de crecimiento limitado, carnosos, cortos y que pueden encontrarse en la superficie (epigeos) o subterráneos. Ejemplos de este tipo de plantas son: colinabo (*Brassica oleracea* L.), que es un tubérculo caulinar epigeo; violeta de los alpes (*Cyclamen persicum* L.), remolacha (*Beta vulgaris*), patata (*Solanum tuberosum*), gladiolos (*Gladiolus spp.*), etc. Algunas plantas tienen tubérculos caulinares llamados pseudobulbos que almacenan agua (ej.: orquídeas terrestres).

- Tubérculo radical: Estructuralmente se parecen a los caulinares pero proceden de engrosamientos de raíces de crecimiento limitado. Ejemplos de este tipo de plantas son la

remolacha azucarera (*Beta vulgaris*), zanahoria (*Daucus carota L.*), nabo (*Brassica rapa*), batata (*Ipomoea batata*), etc.

En las plantas herbáceas encontramos otras modificaciones de los tallos que le permiten arrastrarse o trepar con el objetivo de adaptarse a terrenos irregulares y así aprovechar mejor la luz. Algunos ejemplos son:

a) En las plantas trepadoras (bejucos o lianas): Siempre están arraigadas al suelo:

- Espinas: Son formaciones agudas, afeznadas, ricas en tejido de sostén, rígidas y difíciles de arrancar. La mayoría son proyecciones que se producen por transformación de hojas (espinas foliares), del tallo y ramas (espinas caulinas) o en las raíces (espinas radicales). Las espinas constituyen adaptaciones de las plantas a zonas áridas (cactus), contribuyen a la sujeción de la planta (trepadoras) y como defensa contra herbívoros. Ejemplos de plantas con espinas son: buganvilla (*Bougainvillea glabra*), agracejo (*Berberis vulgaris L.*), espino albar (*Crataegus monogyna J.*), etc.

- Aguijones: Son prolongaciones epidérmicas de los tallos parecidas a pelos engrosados formadas por tejidos corticales y por ende fáciles de arrancar. Por error, suelen confundirse mucho con las espinas. La mayor confusión sucede con los rosales (*Rosa spp.*) que no son espinas sino aguijones.

- Volubles: Son tallos muy largos, delgados y flexibles que tienen la capacidad de enroscarse alrededor de otras plantas o "guías". Los giros pueden ser a la izquierda o la derecha. Las plantas con este tipo de tallos se llaman volubles. Un ejemplo de planta voluble es la judía (*Phaseolus vulgaris L.*).

- Zarcillos: Son estructuras que surgen a partir de tallos (zarcillos caulinares) u hojas (zarcillos foliares) modificadas. En general son filiformes, enroscados, delgados y actúan como fijadores gracias a su intensa excitabilidad al contacto y capacidad de rodear los soportes que se ofrecen y de este modo fijar los tallos respectivos. Ejemplos de plantas con zarcillos son: leguminosas (*Phaseolus spp.*), capuchina (*Tropaeolum majus L.*), pasiflora (*Pasiflora caerulea L.*), etc.

b) En las plantas rastreras

Además de los tipos anteriores existen los estolones que son tallos débiles, postrados y que se arrastran por el suelo y tienen nudos. En cada nudo se desarrollan nuevos tallos y nuevas raíces. Las plantas que tienen esta estructura se llaman estoloníferas. Ejemplos de plantas con estolones son: la fresa (*Fragaria vesca* L.), tomillo limonero (*Thymus x citriodorus*), etc.

c) Otros tipos de tallos

Filocladios y cladios: En realidad son tallos o ramas verdes "comprimidas" que tienen la apariencia de una hoja pequeña. Estas estructuras pueden producir hojas, flores y frutos. Ejemplos de estas estructuras son el rusco (*Ruscus aculeatus*) y algunas de especies de cactus y orquídeas.

De acuerdo a su altura y hábito de crecimiento las plantas leñosas pueden ser...

- Árboles: Son aquellas plantas que tienen un tallo leñoso, rígido, generalmente de más de 5 metros de altura y muy duro que se le conoce como tronco. La característica fundamental de este tronco es que no se ramifica en la base sino que lo hace a gran altura del suelo en lo que se conoce como copa. La arquitectura de los árboles depende del sistema de ramificación que suele ser muy complejo. Ejemplos de árboles son: abeto (*Abies alba* M.), sauce llorón (*Salix babylonica* L.), cocotero (*Cocos nucifera* L.), buganvilla (*Bougainvillea glabra*), etc.

- Arbustos: Son aquellas plantas que tiene un conjunto apretado de tallos leñosos duros (no se distingue un tallo o tronco principal como en los árboles), generalmente entre 1 y 6 metros de altura y cuya característica fundamental es que comienza a ramificarse en la propia base del tallo a nivel de la tierra. Durante el primer año el tallo principal se desarrolla a muy poca altura del suelo. Durante el segundo año se forman ramas muy vigorosas (casi del mismo grosor de tallo) que sobrepasan al tallo principal. Las ramas que se van formando, generalmente, se curvan o inclinan hacia el suelo. Ejemplos de arbustos son: camelia (*Camelia japónica*), aralia (*Fatsia japonica* T.), abelia (*Abelia x grandiflora*

A.), espirea (*Spiraea spp.*), banderita española (*Lantana camara L.*), lavanda (*Lavandula vera D. C.*), sauzgatillo (*Vitex agnus-castus L.*), etc.

- Matas: Son aquellas plantas que tienen las mismas características de los arbustos pero que tienen una altura inferior al metro. También se les conoce como frútices o arbustos enanos. Ejemplos de este tipo de plantas son: hipérico (*Hypericum calycinum L.*), azalea (*Rhododendron indicum*), brecina (*Calluna vulgaris S.*), rododendro (*Rhododendron ferrugineum L.*), gayuba (*Arctostaphylos uva-ursi S.*), arándano (*Vaccinium myrtillus L.*), etc.

- Sufrútices o subarbustos: Es una forma de tránsito entre hierbas (plantas herbáceas) y arbustos (plantas leñosas). Son plantas en las que la parte basal (inferior) del tallo está lignificada (leñosa), mientras que las superiores son herbáceas. Ejemplo de este tipo de plantas son: salvia (*Salvia officinalis L.*), espliego (*Lavandula spica L.*), alhucema (*Lavandula latifolia W.*), etc.

- Hierbas con troncos foliares: Por la altura pueden parecer árboles pero en realidad lo que apreciamos que parece un tallo es en realidad un tronco foliar o falso tallo constituido únicamente por jugosas vainas foliares arrolladas unas alrededor de otras y provistas de tejidos mecánicos. En este grupo destacan: vedegambre (*Veratum album L.*), plátano o banano (*Musa spp.*), etc.

Si nos referimos a la caída de las hojas (follaje) podemos encontrar....

- Plantas de hoja perenne: Son aquellas plantas que, con la llegada del invierno, reducen al máximo sus funciones vitales pero siempre tienen hojas verdes cubriendo sus ramas. Ejemplos de este tipo de plantas son: buganvilla (*Bougainvillea glabra*), cycas (*Cyca revoluta B.*), camelia (*Camelia japónica*), lavanda (*Lavandula vera D. C.*), etc.

- Plantas de hoja caduca: Son aquellas plantas que, con la llegada del invierno, pierden las hojas como respuesta a la drástica reducción del agua disponible. Ejemplo de este tipo de plantas son: granado (*Punica granatum L.*), sauce llorón (*Salix babilonica L.*), árbol de

Júpiter (*Lagerstroemia indica* L.), espirea (*Spiraea spp.*), forsitia (*Forsythia x intermedia*), etc.

- **Plantas de hoja marcescente:** Es un estado intermedio entre la hoja perenne y caduca. Por ejemplo en los robles (*Quercus spp.*), al llegar el otoño, las hojas entran en fase de senescencia con normalidad pasando a ser de color marrón pero no se produce la abscisión, la hoja no cae, permanece en la rama sobre la que se seca, dando al árbol aspecto de estar enfermo o de haberse secado completo. El árbol pasa todo el invierno cubierto con sus propias hojas secas que solo se van cayendo al llegar la primavera a medida que van brotando las nuevas. Se cree que la marcescencia responde a una adaptación al frío, como mecanismo para proteger las yemas contra las heladas.



**MUCRONADO:** Con una punta corta, angosta y rígida en el ápice.

**ACUMINADO:** Provisto de una prolongación estrecha en el extremo de una lámina.

**AGUDO:** Terminado en ángulo agudo.

**EMARGINADO (A):** Con una muesca o talladura poco profunda en el ápice; la mayoría de las veces tratándose de hojas.

**LACINIADO:** Margen dividido irregular y profundamente en dientes angostos y diferentes.

**OBTUSO (A):** Terminado en punta roma o ángulo obtuso.

**GLABRO (A):** Desprovista de pelos o pelusas.

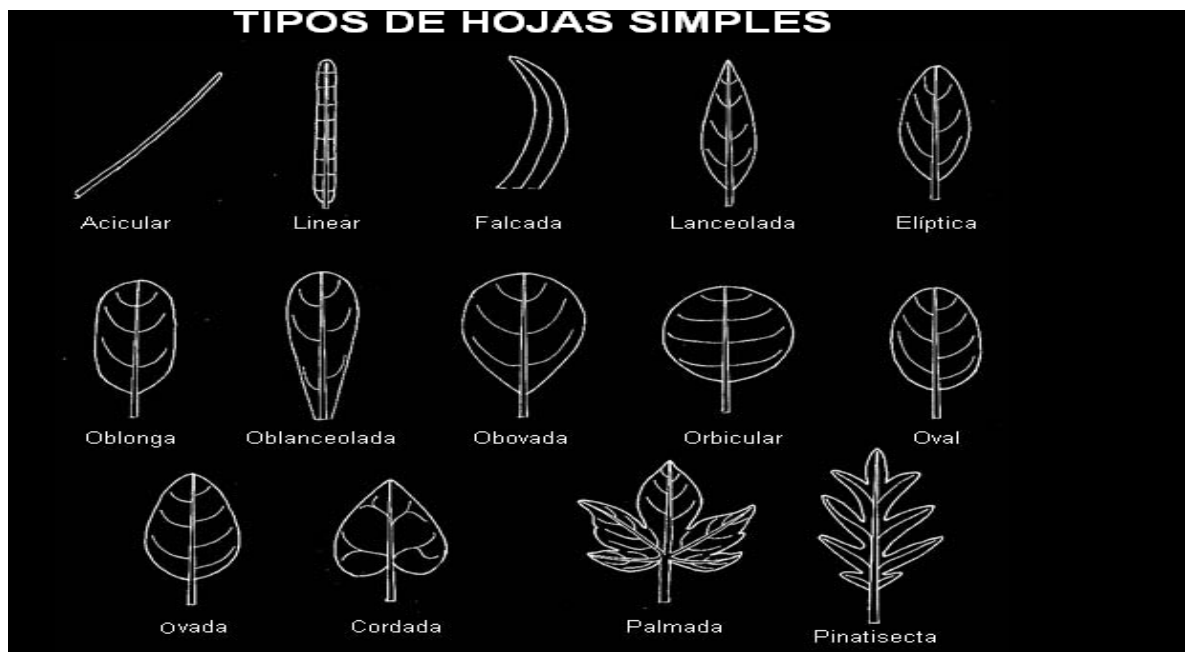
**ESCAMOSO (A):** Con escamas.

**ESCAMIFORME:** Con forma o aspecto de escama.

**GLÁNDULA:** Célula o conjunto de células secretoras.

**GLANDULAR:** Cubierto por pequeños pelos (glándulas) que presentan células secretoras.

**SIMPLE:** No dividido o lobulado.



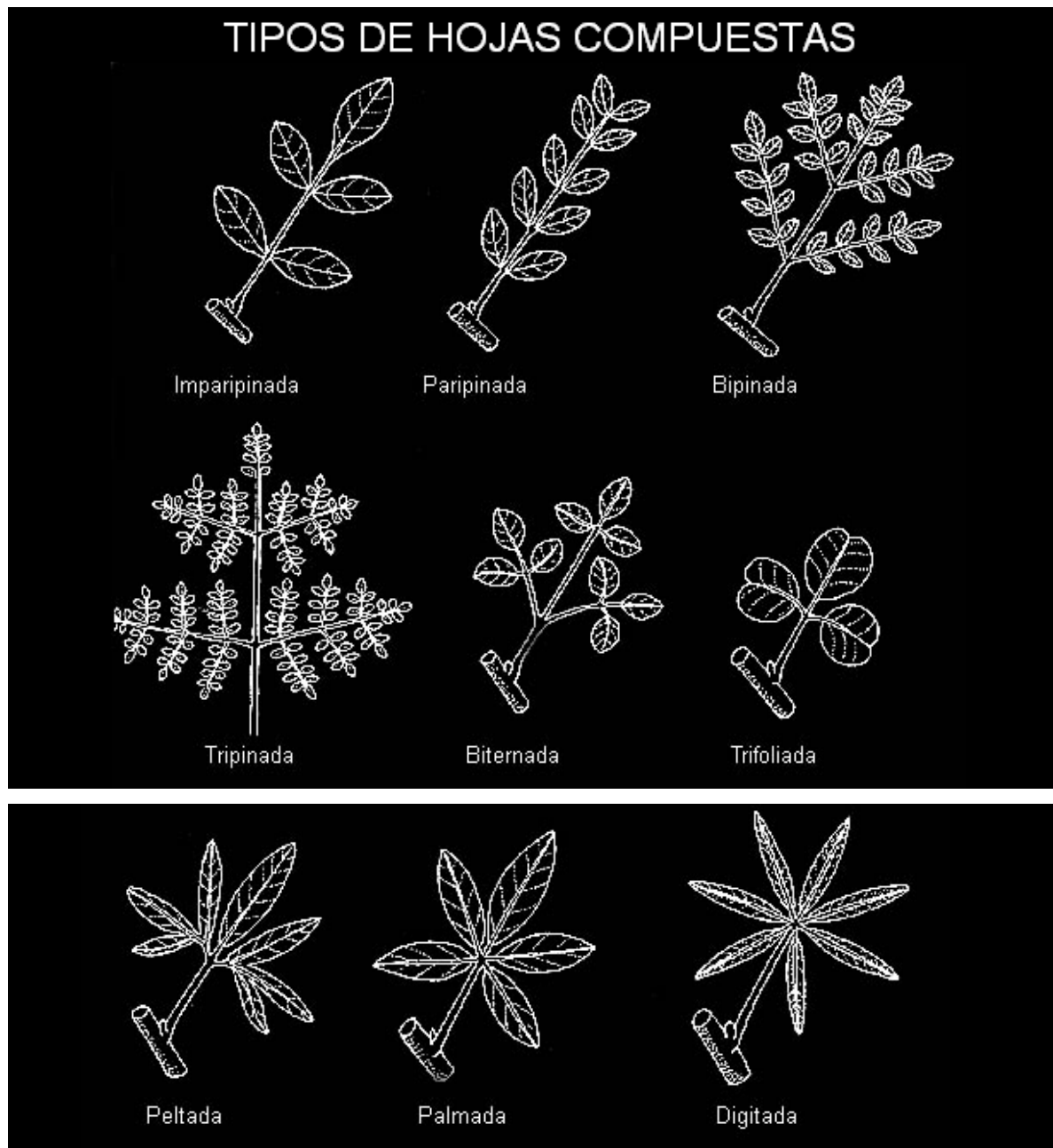
**LANCEOLADO (A):** Órganos laminares, que tienen forma de lanza.

**OVADO (A):** Con forma de huevo, con la base más amplia que el ápice.

**ELÍPTICO (A):** Casi dos veces más ancho que largo, estrechándose de igual forma hacia ambos extremos.

**OBOVADO (A):** Contorno con forma de huevo, pero con la parte más estrecha hacia el punto de fijación.

**PALMADA:** Hoja compuesta por más de 3 folíolos que surgen del mismo punto.

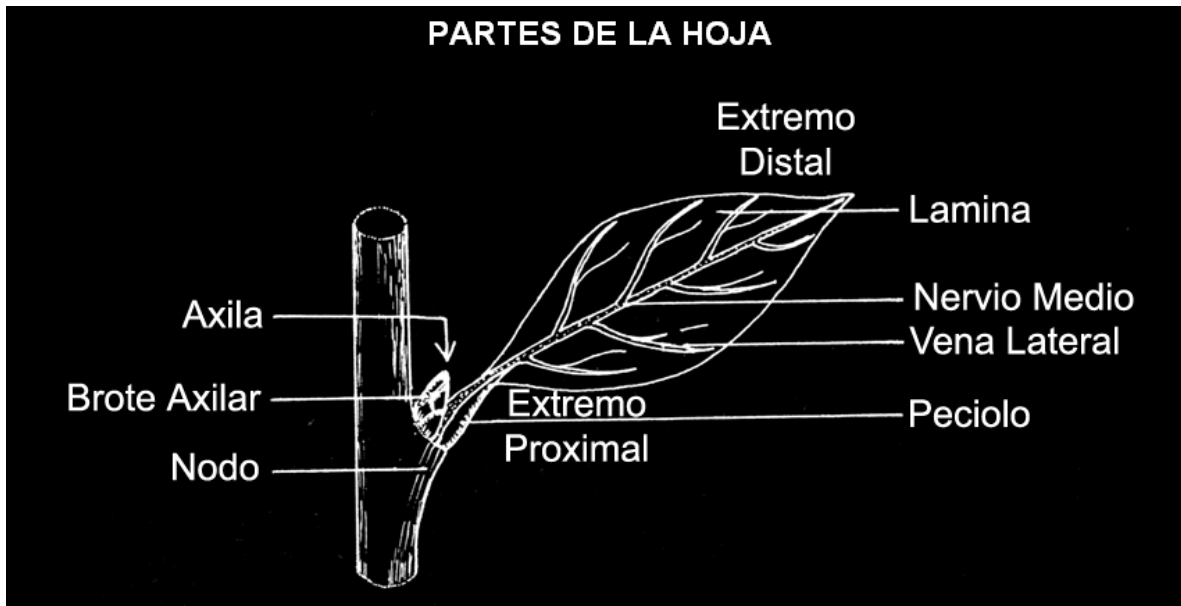


**PINNAS:** La división primaria de una hoja pinnada.  
**PINNADO (A):** Hoja con folíolos distribuidos en ambos lados del raquis.

**FOLÍOLO:** Cada una de las partes finales en que se divide una hoja compuesta.

**ZARCILLO:** Órgano filamentosos que las plantas utilizan para trepar.

**PECÍOLO:** Raballo que une la lámina de la hoja al tallo.



**ESTÍPULA:** Apéndice que se halla en la base de algunas hojas.

**CONCRESCENTES:** Órganos o partes de ellos que pudiendo estar separados están congénitamente unidos.

**DIOICO (A):** Plantas que presentan las flores masculinas en un individuo y las femeninas en otro.

**MONOICO:** Plantas que presentan los órganos sexuales en flores distintas, sobre el mismo individuo.

**HERMAFRODITA:** Flor con los dos sexos.

**CÁLIZ:** Parte externa de la flor, usualmente verde y formada por varias divisiones llamadas sépalos, que protegen la yema.

**COROLA:** Estructura que comprende a los pétalos.

**PÉTALO:** Cada una de las partes de la corola.



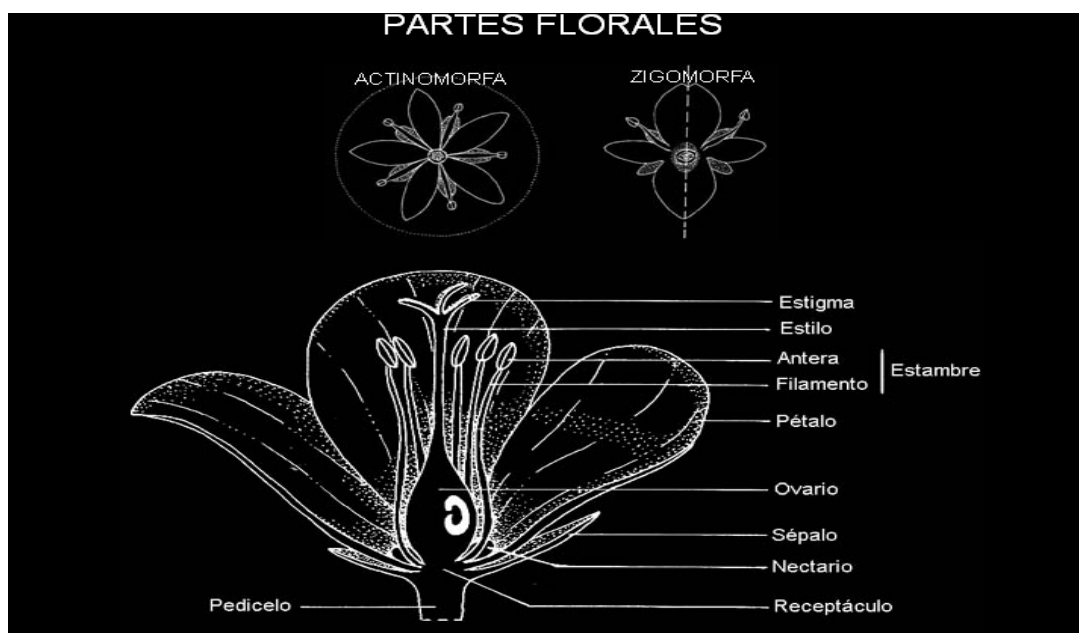
**ESTAMBRE:** Órgano masculino de la flor, que produce polen, generalmente formado por un filamento y una antera.

**ESTIGMA:** Parte del órgano femenino de una flor, en la parte superior del estilo, es el lugar en donde se recibe el polen.

**OVARIO:** Parte del órgano femenino de una flor en donde se ubican los óvulos que finalmente darán origen a las semillas.

**ESTIGMA:** Parte del estilo adecuado para retener el polen.

**CARPELO.** Estructura femenina que contiene los óvulos y soporta al estigma.



**ESPORA:** Corpúsculo reproductor de las plantas criptógamas (como los helechos).

**ESPORANGIO:** Cualquier recipiente en que se contengan esporas.

**INFLORESCENCIA:** Conjunto de flores, las cuales se encuentran lo suficientemente cercanas para formar una unidad estructurada.

**DEHISCENTE:** Manera en que se abre un fruto capsular o antera por valvas o poros, con el fin de liberar su contenido.

**CIMA:** Inflorescencia en la cual la flor terminal se abre primero y las demás flores nacen sobre ramas que se originan desde abajo.

**CIMOSO (A):** Que soporta cimas o relacionado a una cima.

# EXPLICACIÓN DE LA HERRAMIENTA NITIDA

Una vez obtenidas estas características realizamos un primer programa. Este programa obtiene de un fichero de texto el total de las familias de la flora ibérica “plantas.txt” y según el usuario va marcando las características de la planta a determinar se van eliminando las familias que nos las cumplan.

También podemos ver las características de cada una de ellas así como las características por las que han sido eliminadas las familias descartadas.

Ahora explicare los métodos más importantes del programa inicial que hemos realizado.

La clase principal es Princip.java. En ella se inicializa la variable Fich que es de la clase Ficheros.java con el fichero de texto plantas.txt.

Una variable Familias como un ArrayList y después se inicializa las componentes graficas de este JFrame, una ventana.

También se inicializa una variable p del tipo PanelPrin que lo que hace es crear un nuevo panel dentro de este.

En el método initComponents se puede inicializar las componentes gráficas de esta ventana.

Ponemos dos menús:

Uno de ellos es Archivo con las opciones Abrir, Guardar y Cerrar.

Otro de esos menús es Operaciones con las opciones Comenzar, Reiniciar y Descartados.

Cuando pulsamos la opción Cerrar simplemente se cierra la herramienta.

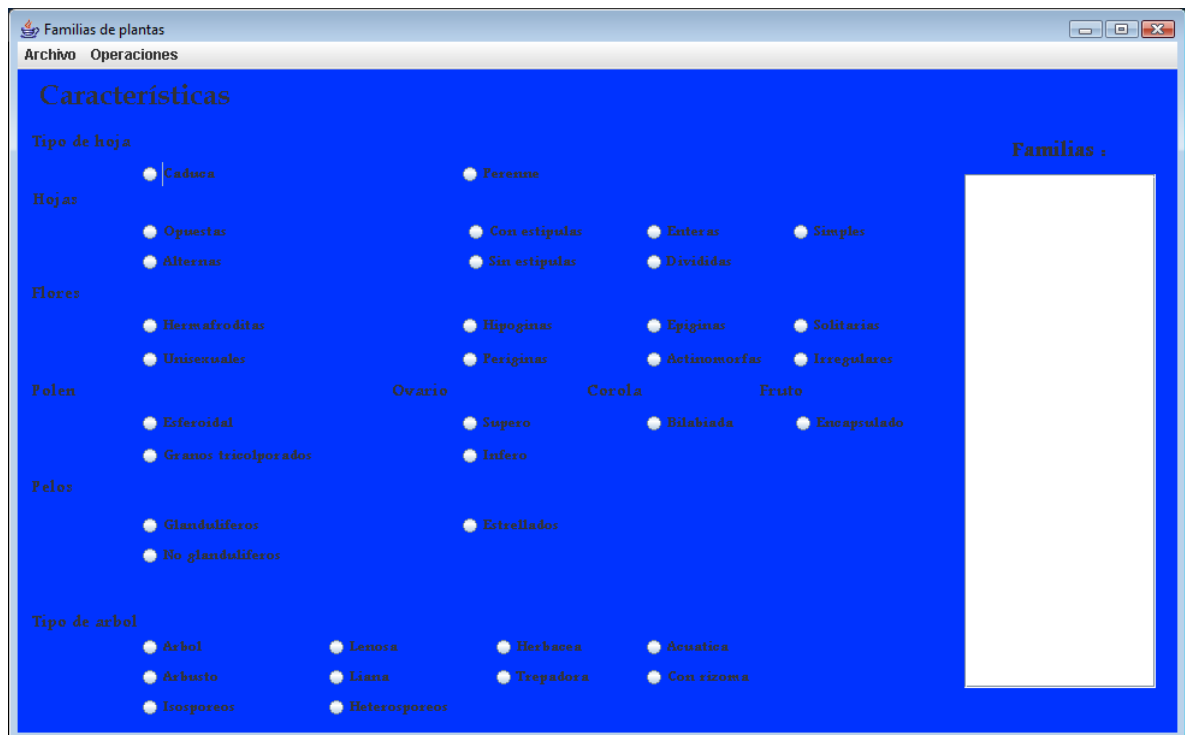
Cuando pulsamos la opción de Descartados creamos una variable de la clase Descartados.java llamada desc.

Seguidamente ejecutamos los métodos Poner de esta clase, Inicializa y hace que la ventana sea visible para el usuario.

La opción Reiniciar ejecuta el método Reiniciar de la clase PanelPrin.java.

Cuando se pulsa la opción Comenzar lo que hacemos es ejecutar el método leer de la clase Ficheros.java en el que anteriormente se ha inicializado con el fichero plantas.txt y además ejecuta el método Poner de la clase PanelPrin.java al que se le pasa el ArrayList

Familias que lo que tiene después de haber ejecutado el método leer de la clase Ficheros.java es el nombre de la familia seguida de sus características en cada posición del array. Estos dos métodos los explicaremos en sus respectivas clases.



En esta imagen se puede ver las opciones descritas en la clase Princip.java de los menús Archivo y Opciones.

Ahora explicaremos la clase PanelPrin.java

En ella ponemos las características explicadas anteriormente.

En esta clase lo primero que realizamos es inicializar el fichero plantas.txt mediante la clase Ficheros.java.

Esta clase se describirá mas tarde.

Se inicializan dos ArrayList uno para las familias y otro para los nombres de las familias.

Además se inicializan todos los componentes gráficos de la herramienta. Estos los describimos seguidamente dando una visión de la interfaz usada en dicha herramienta.

Se puede ver como el título de la herramienta es Familias de plantas, con ello queremos decir que estamos realizando una determinación de las plantas dando como resultado su familia. Es decir la familia a la pertenece la planta a determinar.

Después introducimos una serie de JLabels, que son etiquetas que se muestran en el Panel como por ejemplo: Características, Tipo de Hoja, Flores, Tipo de árbol, etc.

Para estos JLabels se puede elegir tanto el tamaño como la ubicación, así como la fuente de la letra, etc. Podemos establecer cualquier característica relativa a la letra.

Para la etiqueta Características simplemente es para mostrarla en el panel. El resto de las etiquetas llevan a continuación de esta las distintas descripciones que puede tener dichas características.

Para cada una de las descripciones de las características introducimos un JLabel, es decir otra etiqueta con cada una de las descripciones de esta y para cada una de ellas ponemos los JRadioButton situado a su izquierda, necesarios para que el usuario señale o no si su planta posee dicha descripción de esa característica.

**Por ejemplo:**

Tipo de Hoja:

Con un JRadioButton para la etiqueta de la descripción de tipo de hoja perenne con el texto Perenne y otra etiqueta con otro JRadioButton para la etiqueta de la descripción de tipo de hoja caduca con el texto Caduca.

Los JRadioButton son botones los cuales cuando los pulsas aparece el círculo que tienen a la izquierda de cada texto pulsado.

Así ponemos 36 botones los cuales pertenecen a la descripción de cada una de las características de las plantas. También incluimos las etiquetas para conocer la descripción de la característica que queremos señalar.

El usuario debe pulsar cada uno de los JradioButton que coincida con la misma descripción de la característica de su planta. Con ello conseguimos conocer las características que posee la planta a determinar por el usuario.

Para conocer las características que posee la planta a determinar debemos de pulsar el JradioButton que esta a la izquierda de la descripción de la característica que sea cierta para esa planta.

Pulsando este JRadioButton conseguimos capturar el evento que se produce con un método llamado MouseClicked es decir con el evento clic del ratón sobre el botón.

Cada JRadioButton posee el mismo evento, cada vez que pulsamos uno de ellos ya no podemos realizar ninguna otra acción sobre este botón, ya que la posibilidad de pulsar sobre el mismo botón se desactiva.

Si se quiere desactivar una vez activado este botón el usuario debe de reiniciar el programa.

Además se ejecuta el método comprobar al que se le pasa un String con el texto de la etiqueta del botón que esta a la izquierda de este.

En la imagen anterior vemos un panel en blanco a la derecha de la imagen. En ella vamos a poner el nombre de las familias, inicialmente estará vacío, después cuando pulsamos la opción del menú Opciones Comenzar se colocaran en este panel en blanco todas las familias. Cuando se van pulsando los botones en este panel solo se quedaran las familias que aun siguen siendo aceptadas con las características que el usuario haya introducido.

El panel que es de tipo Jlist posee un evento que explicaremos a continuación.

En este evento capturamos la familia que esta pulsada y mediante la variable NomFam obtenemos el nombre de la familia pulsada.

Entonces aparecerá una nueva ventana con las características de esta familia. Para esto debemos de crear una variable de la clase Aceptados.java llamada acep para ejecutar los métodos, uno para que esta ventana sea visible y el método Escribir de esta misma clase.

La clase Aceptados.java la explicaremos a continuación.

Primero explicaremos el método Poner para poder comprobar lo que realiza el método comprobar que se ejecuta en cada evento de los botones.

Al método Poner se le pasa un ArrayList desde la clase Princip.java un ArrayList que tiene en cada posición un array cuya primera posición es el nombre de la familia y en posiciones posteriores las características de las que consta la familia. Todo esto es obtenido del fichero plantas.txt leído por el método Leer de Ficheros.java.

Al ArrayList de esta clase (PanelPrin.java) llamado Familias se inicializa copiando el ArrayList obtenido de la lectura del fichero plantas.txt.

Seguidamente se coge el array de cada posición del ArrayList Familias con un bucle de 0 a 115, la longitud de este ArrayList.

De cada posición del ArrayList que es un array cogemos la primera posición que es la que tiene el nombre de la familia y la ponemos en otro array llamado auxNom, el cual al finalizar el bucle tiene todos los nombres de las familias en cada una de sus posiciones.

Este array se le pasa a la componente jList1 que es el cuadro de texto que se muestra en la parte derecha de la pantalla de la herramienta.

Además la variable NomFam se inicializa con auxNom con lo que en dicha variable tenemos todos los nombres de las familias.

En el método comprobar se le pasa un String con el nombre de la característica a comprobar.

Inicializamos el array nombres como se inicializa cualquier ArrayList vacío.

Después realizamos un bucle de 0 a 115, la longitud del ArrayList NomFam es decir 116 posiciones al igual que el número de familias que existen.

De cada una de ellas obtenemos su nombre de cada posición del ArrayList NomFam y con este nombre comprobamos la característica pasada como parámetro del método comprobar. Este método se llama comprobarCarac y será explicada a continuación.

Ahora vamos a actualizar el ArrayList NomFam con las familias que siguen siendo Aceptadas después de haber pulsado uno de los botones y por tanto alguna de las familias que no cumplen esta característica deben de ser descartados de la lista de familias aceptadas que es la que se ve en el cuadro de texto de la pantalla principal a su derecha.

El ArrayList NomFam se actualiza al tamaño del ArrayList nombres y de cada una de sus posiciones obtenemos el nombre de la familia y lo introducimos en el ArrayList NomFam quedando este con los nombres de las familias que siguen siendo aceptadas una vez que se pulsa una de las características, uno de los botones de la herramienta.

Cuando el bucle se finaliza se actualiza el elemento grafico del cuadro de texto con las familias que siguen siendo aceptadas ósea se le pasa el ArrayList NomFam que es el ArrayList que tiene esto.

Después vemos el método comprobarCarac al que se le pasa como parámetros el nombre de la familia que se quiere comprobar que tiene esta característica, el String con el texto de la característica que se quiere comprobar y el ArrayList Familias que tiene todos

los nombres de las familias en cada posición del ArrayList además de sus características en sus posiciones posteriores.

Realizamos un bucle de 116 reiteraciones obteniendo en el objeto ob cada posición del ArrayList Familias.

En la variable nom obtenemos el nombre de la familia cogiendo del objeto ob su primera componente que es en efecto el nombre de cada familia.

Comprobamos si el nombre pasado como parámetro coincide con este, y cuando esto se cumple tenemos que comprobar que la característica pasada como argumento la posee la familia que también esta pasada como argumento.

Esto se consigue mediante el método comprobar2 que explicaremos a continuación.

Este método devuelve un booleano. Si es false la característica no se cumple para la familia, en caso de ser true se cumple por lo tanto se actualiza el ArrayList nombres añadiendo esta familia ya que en este ArrayList van a quedar las familias que cumplen la característica que esta pasada como argumento y que hemos obtenido mediante el evento del botón pulsado cogiendo el texto del JLabel que le acompaña, es decir, la característica que queremos que las familias mostradas como aceptadas tengan en su descripción.

El método comprobar2 le pasamos como argumento la característica a comprobar y el ArrayList que tiene tanto el nombre de la familia como sus características en las posiciones siguientes.

Como lo que queremos comprobar es si la característica esta en el array de características que cumple la familia. El ArrayList aux tiene en su primera posición el nombre de la familia y en sus siguientes posiciones las características que cumple la familia.

Por lo tanto hacemos un bucle desde 0 hasta el tamaño de aux comprobando cogiendo una por una las características del array aux y comprobando que es igual que las características que queremos comprobar y que le hemos pasado como parámetro al método.

Si es igual este método devuelve true y si ninguna de las características de la familia cumple la pasada como argumento este método devuelve false.

El método que le sigue es Reiniciar el cual en el ArrayList NomFam se inicializa a vacío es decir que el cuadro de texto que aparece en la pantalla en la parte derecha de esta, no se muestra ninguna familia.

Además todos los botones se deselectan y permitimos seleccionarlos de nuevo.

Este método reinicia la herramienta.

El método `verSeleccionados` devuelve un `ArrayList` con las características que han sido señaladas en la herramienta.

Esto se consigue mirando si los botones están seleccionados o no y si lo están añadimos a este `ArrayList` el texto del botón que es la descripción de la característica así como la característica pero esto negado.

Es decir, Tipo de hoja no caduca. Tipo de hoja es la característica y la descripción caduca, ahora se niega, ósea, no caduca. Esto es debido a que este método es llamado por la clase `Descartados.java` cuando se pulsa en la opción Razones descartados y por lo tanto tenemos que conocer los botones pulsados para conocer las razones por las que han sido descartadas las familias de esta lista.

La clase `Aceptados.java` se inicializan todos sus componentes al ser un panel. El título de este panel es características de la familia.

Después se añade un componente gráfico llamado `JTextArea1` en el que se van a escribir todas las características de la familia seleccionada.

También tiene un botón con el texto Aceptar cuyo evento explicaremos más tarde.

Añadimos dos etiquetas una que pone Nombre de la Familia y otra que pone características.

Cuando se pulsa el botón Aceptar se ejecuta el evento que es que este panel no sea visible. Entonces parecerá que esta ventana se ha cerrado.

Luego tenemos el método `Escribir` al que se le pasa el nombre de la familia como parámetro.

Después leemos mediante el método `Leer` de la clase `Ficheros.java` leemos el fichero `plantas2.txt` que tiene todas las características de las familias además de las que puede seleccionar el usuario todas las que la familia posee.

Ahora recorreremos el `ArrayList` que se obtiene de la lectura del fichero `plantas2.txt` mediante un bucle de 116 reiteraciones cogiendo el nombre de la familia que está en la primera posición de cada uno de los elementos del `ArrayList` contiene el nombre de la familia.



Si el nombre de la familia pasado como argumento o parámetro al método coincide la clase que en ese momento se esta procesando se ejecuta el método Escr al que se le pasa el nombre de la familia.

En el Escr se coge una a una las características de la familia.

Están en el ArrayList Fami, en su primera posición tiene el nombre de la familia y en las sucesivas todas las características de esta familia, obtenidas del fichero plantas2.txt.

En este método se van cogiendo cada uno de los String de cada una de las posiciones del ArrayList Fami excepto la primera que es la que tiene el nombre de la familia.

Cuando leemos un String que termina en dos puntos, es decir “:”, se pone el texto es la característica y se pone en el JTextArea1 seguido de los dos puntos.

Si antes de los puntos hay un punto quiere decir que la característica consta de más de una palabra y si no aparece el punto quiere decir que la característica consta de una sola palabra.

Además después se pone una línea de separación, es decir, se hace un retorno de carro para que lo siguiente se muestre en la línea siguiente a esta.

Si leemos un punto añadimos al principio de la línea espacios en blanco y después la descripción de la característica anteriormente leída. Todas las descripciones de la característica vienen seguidas de un punto.

La clase Descartados.java es un panel que tiene como titulo Descartados y que posee un menú de opciones con la etiqueta Menú y otro menú con la etiqueta Opciones que tiene como opciones Características, Razones descartadas y Cerrar.

El menú de opciones Características tiene el evento que ejecuta lo siguiente.

Crea un nuevo objeto de la clase Aceptados.java y ejecuta los métodos para que se muestre la ventana y el método Escribir en el que se muestran las características de la familia que se ha señalado previamente de pulsar esta opción en el menú de opciones Opciones.

En la opción Razones descartadas creamos un nuevo ArrayList res en el que se guardan mediante el método verSeleccionados de la clase PanelPrin.java todas las razones por las que esta descartada la familia que se le pasa como argumento a este método.

También se hace visible el panel de la clase Porque.java que explicaremos a continuación.

Y sobre dicho panel se ejecuta el método Poner de esa misma clase pasándole como argumento el nombre de la familia y el ArrayList res con las razones por las que esta descartada esta familia.

Además también se incluye como elemento grafico un JList1 en el que se ponen todos los nombres de las familias que han sido descartadas gracias a las características que el usuario ha introducido en la pantalla principal de la herramienta.

El evento de este elemento grafico podemos conocer el nombre de la familia que ha sido señalada para a continuación poder ejecutar las opciones del menú de Opciones tanto Características como Razones descartadas. Ya que estos métodos necesitan previamente conocer el nombre de la familia para ejecutarse.

El nombre de la familia se guarda en la variable nombre a la que pueden acceder todos los métodos de esta clase.

En el método Poner se le pasa como parámetro un ArrayList F que tiene el nombre de todas las familias ya que se lo pasamos desde el evento del menú de Opciones Descartadas de la clase PanelPrin.java. Además también se le pasa un array de String que tiene el nombre de las familias aceptadas en ese momento de la ejecución del programa mediante la variable NomFam de la clase PanelPrin.java.

Teniendo el nombre de todas las familias y el nombre de las familias aceptadas hasta el momento podemos sacar el nombre de las familias descartadas.

Para ello recorremos todo el ArrayList Familias que se ha inicializado con el ArrayList F pasado como argumento.

En el momento en el que comparamos cada una de las componentes de este ArrayList con el array NomFam que tiene el nombre de las familias aceptadas hasta ese momento e inicializado con el parámetro NomF pasado como argumento y coincide quiere decir que esa familia esta aceptada y por tanto pasamos a comprobar la siguiente familia.

Cuando no coincida entonces añadimos al array aux esa familia porque quiere decir que esa familia no está entre las aceptadas.

Después de ejecutar todo el bucle pasamos al componente grafico JList1 este array para que se escriban todos los nombres de las familias descartadas en ese elemento.

La clase Porque.java muestra un panel que tiene como título Razones descartadas, ya que esta clase se ejecuta cuando se pulsa en el menú de opciones de Razones descartadas de la clase Descartados.java.

Se añaden dos etiquetas una con el texto Nombre de la familia y otro con el texto Porque ha sido descartado.

Además añadimos un botón con el texto Aceptar cuyo evento es que el panel no sea visible. Entonces parecerá como si se hubiese cerrado la ventana.

Tiene un método llamado Escribir al que se le pasa como argumento el nombre de la familia que ha sido señalada en la lista que aparece de familias descartadas en el panel de la clase Descartados.java. También se le pasa como argumento el ArrayList obtenido de ejecutar el método verSeleccionados de la clase PanelPrin.java que lo que tiene es lo explicado para este método anteriormente.

Entonces lo que realiza este método es escribir en un JTextArea1 lo que tiene el ArrayList pasado como argumento llamado pq. Introduciendo después de cada una de estas componentes un retorno de carro para que cada una de las componentes se muestre en una línea de este elemento grafico (JTextArea1).

La clase de Ficheros.java tiene una constructora a la que se le pasa un parámetro String que es la ruta del fichero que queremos leer.

Entonces creamos una variable ArrayList llamada familias para guardar todo lo que leamos en el fichero introducido como parámetro cuando se inicializa dicha clase desde una clase exterior.

Vemos si la ruta del fichero es la correcta, es decir, si el fichero que se le pasa como parte final de la ruta pasada como parámetro existe dentro de la ruta que se indica.

El método Leer inicializa el ArrayList familias como se inicializa cualquier ArrayList vacío. Además lee la primera línea guardándola en una variable tipo String llamada línea, si esta variable no es igual a null se ejecuta el bucle hasta que se llegue al final del fichero ósea hasta que este variable sea igual a null.

Este bucle se ejecuta hasta el final del fichero y ejecuta el método TratarLinea al que se le pasa como parámetro la variable línea que después explicaremos lo que realiza.

Además se lee la siguiente línea del fichero y se le asigna a la variable línea. Si esta es igual a null el bucle finaliza ya que el fichero a leer ya no posee mas líneas y sino sigue con el mismo proceso.

El método Leer posee un tratamiento de error que cuando la ruta pasada como parámetro al método no existe.

Devuelve el ArrayList Familias que se inicializa en el método TratarLinea.

El método TratarLinea se le pasa como parámetro un String.

Para tratar la línea inicialmente vemos que el String tiene más de una silaba. Después realizamos un bucle hasta el final del String, mirando si hay un espacio en blanco se añade a un ArrayList de este método llamado aux una variable de tipo String llamada nom.

Esta variable coge valores cuando estamos en el bucle y leemos algo que no es un espacio en blanco por lo tanto será un símbolo del alfabeto. Cada vez que se siga sin leer un espacio en blanco se añade a la variable nom ese símbolo leído. Formando de esta manera una palabra.

Cuando se lee el espacio en blanco la palabra ha terminado de coger todos sus símbolos que la forman y por tanto la variable nom ya que puede añadir al ArrayList carac que es el que va a tener todas las palabras de la línea.

Es decir, en su primera posición el nombre de la familia y en las sucesivas las características de esta.

Cuando se llega al penúltimo símbolo del String la variable nom se añade al ArrayList carac debido a que el String finaliza ya.

Después de ejecutar el bucle ya tenemos en el ArrayList carac todas las características de la familia que tenemos en la primera posición de este ArrayList.

Entonces lo copiamos a la variable ArrayList familias que es lo que el método Leer explicado anteriormente devuelve.

Ahora vamos a explicar los dos ficheros que este programa usa.

El primero es plantas.txt en el se muestra en cada línea inicialmente el nombre de la familia y después cada una de las descripciones de las características de la familia que esta en la primera posición.

Las descripciones tienen que coincidir con el texto que hay en las etiquetas que se muestran en la pantalla principal de la herramienta.

Como ejemplo en la etiqueta que pone Caduca, en el fichero este debemos de añadir la palabra Caduca si la familia que se encuentra en el inicio de la línea posee esta característica.

En cada línea ponemos como hemos dicho anteriormente el nombre de familia, después un espacio en blanco, seguido de una de las características que cumple la familia, a esta también le sigue un espacio en blanco y así con todas las que posea la familia.

Una vez finalizada la introducción de características de la familia se pulsa intro, retorno de carro, para introducir la siguiente familia realizando esto para todas las familias de la flora ibérica.

La descripción de cada una de las características debe de coincidir con el texto que de ella pone en la pantalla principal de la herramienta. Por ello se pueden poner solamente en este documento las características que se ven en la pantalla y no más.

**Por ejemplo:**

```
Acanthaceae      Perenne      Opuestas      SinEstipulas      Simples
Hermafroditas    Solitarias   Irregulares    Supero      Encapsulado
Glanduliferos    NoGlanduliferos Arbusto
```

El fichero plantas2.txt colocamos en primer lugar de cada línea el nombre de la familia. Después introducimos un espacio en blanco. Seguidamente ponemos la característica que primero queramos. Si esta característica consta de mas de una palabra antes de introducir los dos puntos que siempre la siguen ponemos un punto.

**Por ejemplo:**

Tipo de Hoja.:

Al constar de más de una palabra la característica debemos de poner la característica seguida de un punto y dos puntos después.

Si la característica posee una sola palabra no es necesario introducir un punto, sino simplemente dos puntos.

**Por ejemplo:**

Pelos: Glanduliferos.

Como consta de una sola palabra la característica va seguida solamente de dos puntos.

Cada una de las descripciones de las características conste o no de varias palabras siempre va seguida de un punto y de un espacio en blanco antes y después de escribirla.

Tipo de Hoja.: Caduca. Perenne.

Como se puede ver antes de la característica y después del punto que va detrás de ella se debe de poner un punto.

Lo de las mayúsculas solo se pone para que luego cuando lo mostramos quede más visible,

Así para cada una de las características y para todas sus descripciones de esta.

Este proceso se debe de hacer para todas y cada una de las familias de las que consta la flora ibérica.

Ahora voy a explicar lo que se puede realizar en este programa.

La clase que se ejecuta desde el fichero nítido.bat ejecuta la clase Princip.java y aparece una pantalla con los menús de opciones Archivo y Opciones así como el título de la herramienta.

Además añadimos la ventana que esta implementada en la clase PanelPrin.java que consta de todas las etiquetas, todos los botones y el cuadro de texto en el que se van a introducir las familias.

Las familias se cogen del fichero plantas.txt, esta va a ser su ruta de datos ya que se encuentra en la misma carpeta que el proyecto de eclipse.

Para empezar a ejecutar el programa debemos de pulsar en el menú de Opciones la pestaña que pone Comenzar.

Cuando pulsamos esta opción se ejecuta el evento de esta pestaña, que lo que hace es leer del archivo plantas.txt todas las familias con sus características además también se

ejecuta el método Poner de la clase PanelPrin.java que lo que hace es coger cada una de las familias y las coloca en el cuadro de texto que se muestra en la parte derecha de esta pantalla.

Cuando pulsamos alguno de los botones lo que ejecuta el evento de cada botón es la comprobación de la característica en cada una de las familias de la flora ibérica.

Cogiendo el texto de la etiqueta que aparece en la parte derecha del botón pulsado comprobamos en el array de las familias, que en cada posición del array tiene un nuevo array en el que en la primera posición tiene el nombre de la familia y las siguientes sus características, que la característica la posee o no la familia.

Esto lo realiza el método comprobar. Lo que resulta de su ejecución es la actualización de las familias en el cuadro de texto de la derecha de la pantalla. Ahora ya solo se mostraran las familias que poseen entre sus características la característica pulsada en el botón.

También se pueden ver las características que poseen las familias que están en el cuadro de texto. Basta para verlas pulsar en el nombre de la familia y entonces mediante el evento del cuadro de texto conocemos el nombre de la familia pulsada y creamos un nuevo objeto de la clase Aceptados.java que lo que consigue es mostrar una ventana en la que se muestran las características de la familia pulsada. Esto se consigue ejecutando el método Escribir de esa clase que lee el fichero plantas2.txt en el que están todas las características de las plantas, las que están en la pantalla y las que no aparecen en ella.

Para cerrar esta ventana basta con pulsar el botón Aceptar para que la ejecución del programa sea normal.

Cuando pulsamos alguno de los botones no se puede deseleccionar para ello tenemos una opción de Reiniciar que lo que hace es deseleccionar todos los botones pulsados y el cuadro de texto con las familias aparece vacío como al principio de la ejecución del programa.

Después de pulsar alguno de los botones de la pantalla principal la lista de las familias del cuadro de texto se actualiza y alguna de ellas se elimina.

Cuando alguna de las familias ha sido descartada si pulsamos la opción del menú de Opciones, Descartados aparece una nueva ventana en la que vemos un menú y una lista de las familias que han sido rechazadas.

Cuando pulsamos una de las familias el evento del cuadro de texto la captura para poder pasársela a cada una de las opciones del menú Opciones.

En el menú de Opciones se puede pulsar la opción Características, entonces mediante la clase Aceptados.java se pueden ver las características de la familia que previamente ha sido pulsada.

Las características que se pueden ver en esta ventana son además de las que podemos pulsar en la pantalla principal, todas las características que posee la familia pulsada mediante la lectura del fichero plantas2.txt.

Para cerrar esta ventana basta con pulsar el botón Aceptar para cerrar correctamente esta ventana.

Cuando pulsamos en el menú de Opciones la pestaña Razones descartadas se ejecuta el método verSeleccionados de la clase PanelPrin.java que lo que tiene es un array de Strings con las razones por las que ha sido descartada la familia que previamente ha sido seleccionada. Además se crea un objeto de la clase Porque.java que lo que hace es sacar una ventana en la que aparecen las razones por las que ha sido descartada esa familia.

Como hemos dicho anteriormente esto se obtiene mediante el método verSeleccionados que lo que hace es comprobar cada uno de los botones y ver si están seleccionados o no. Si están seleccionados se pone que no cumple esa característica la familia pulsada.

Cuando pulsamos en el menú de Opciones la pestaña Cerrar entonces se cerrara correctamente la ventana que mostramos mediante la opción de la pantalla principal Descartados que ejecuta la clase Descartados.java.

Podemos pulsar tantos botones como queramos en la pantalla principal además de poder reiniciar la herramienta el número de veces que desee.

También se pueden ver las características de las familias que están en el cuadro de texto de la derecha de la pantalla principal.



Además se pueden ver las familias que han sido descartadas por no poseer alguna de las características que han sido seleccionadas en la pantalla principal. En estas familias se pueden ver sus características además de las razones por las que ha sido descartada.

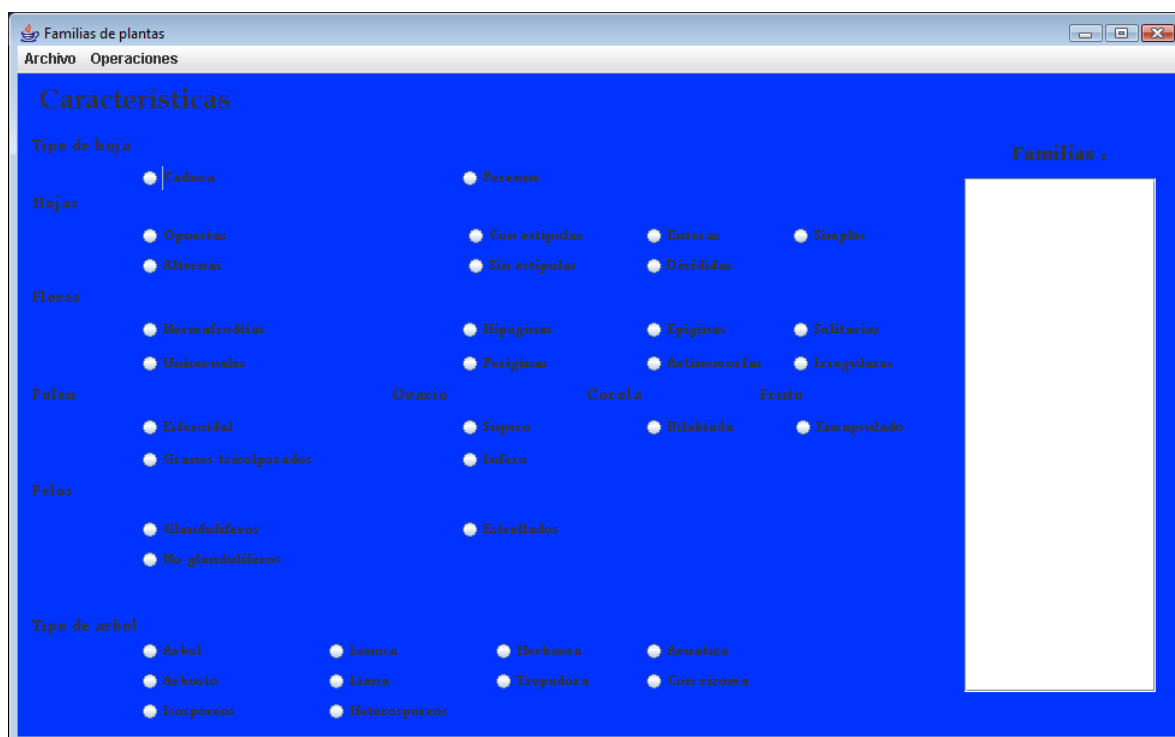
El usuario puede pulsar tantos botones como desee, entonces se mostraran una o varias familias entre las que es posible que la planta que el usuario esta determinando este dentro de dicha o dichas familias.

Así el usuario obtendrá una respuesta a la determinación de su planta, novedad en la que basamos nuestra herramienta ya que siempre proporcionaremos un resultado de la determinación. Por ello creemos que esta herramienta será de gran utilidad a la hora de que un usuario quiera conocer la familia a la que pertenece la planta que quiere determinar.

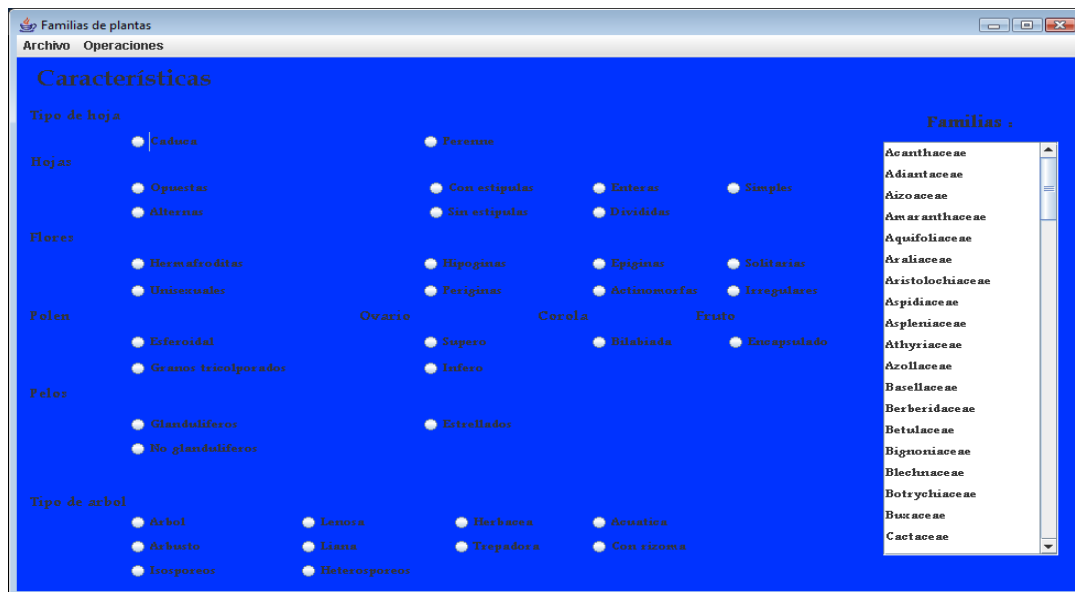
## EJECUCIÓN DEL PROGRAMA

Para ejecutar el programa es necesario hacer doble clic en el fichero.bat que hemos generado para dicho fin.

Una vez hecho esto se mostrara la pantalla principal del programa que será como la siguiente figura:



Entonces para empezar a determinar las plantas, debemos de pulsar al menú Opciones y seleccionar la opción Comenzar. De esta forma aparecerán las familias en el cuadro de texto mostrado en la anterior imagen en la parte derecha. Como la siguiente figura:



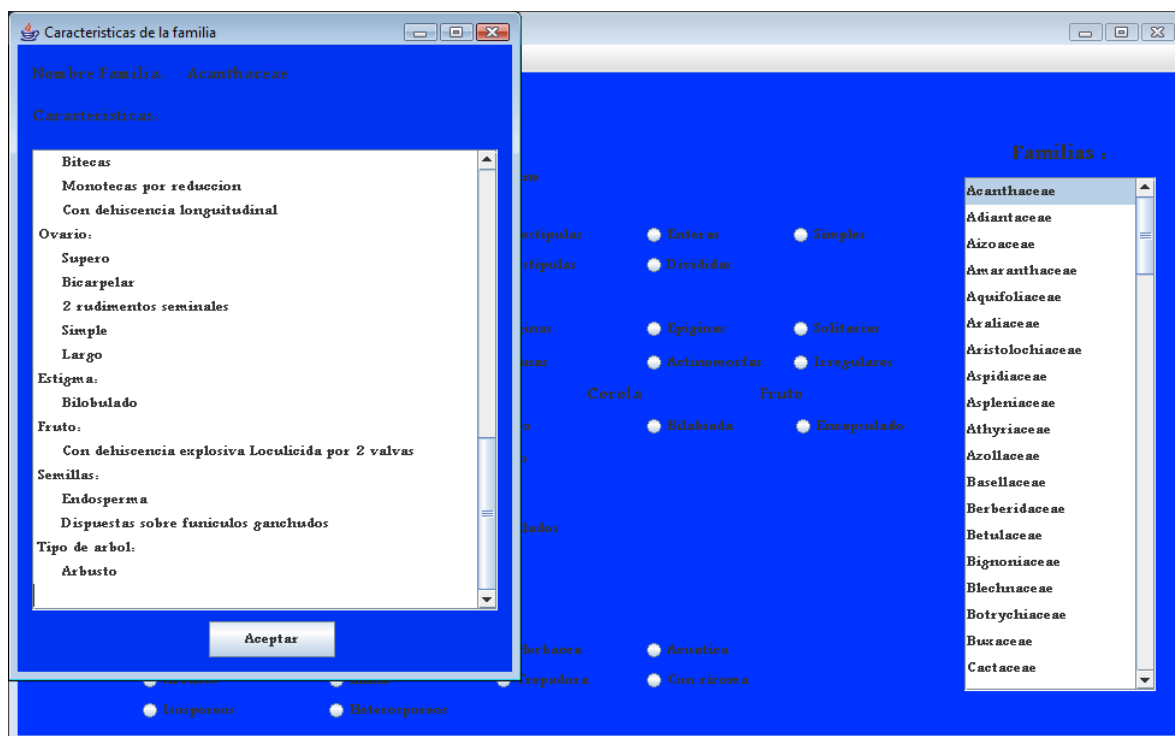
Después podemos comenzar a pulsar cualquier botón mostrado en la pantalla, consiguiendo así eliminar familias de la lista pudiendo obtener la familia a la que pertenece la planta a determinar.

Cuando vas pulsando en los botones se muestran como la siguiente figura:



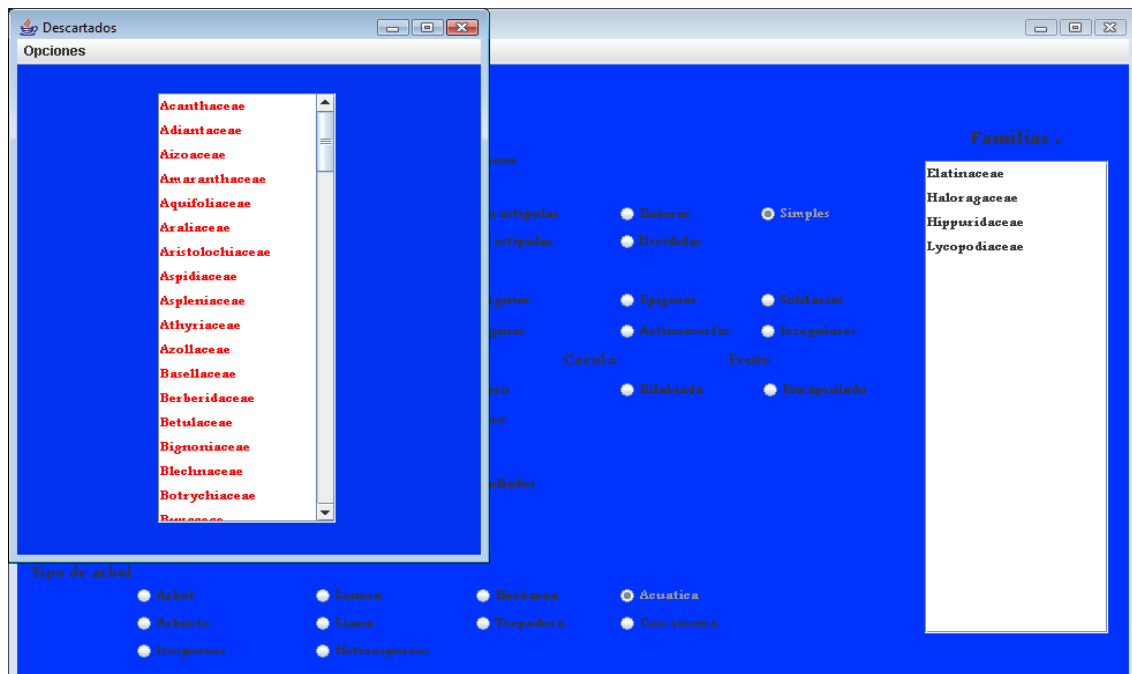
En la parte derecha solo se muestran las familias que siguen siendo aceptadas una vez señaladas las características que se muestran. Que son hermafroditas, simples y acuáticas.

Para ver las características de la familia, debemos de pulsar en una de ellas y entonces saldrá una pantalla como la siguiente:



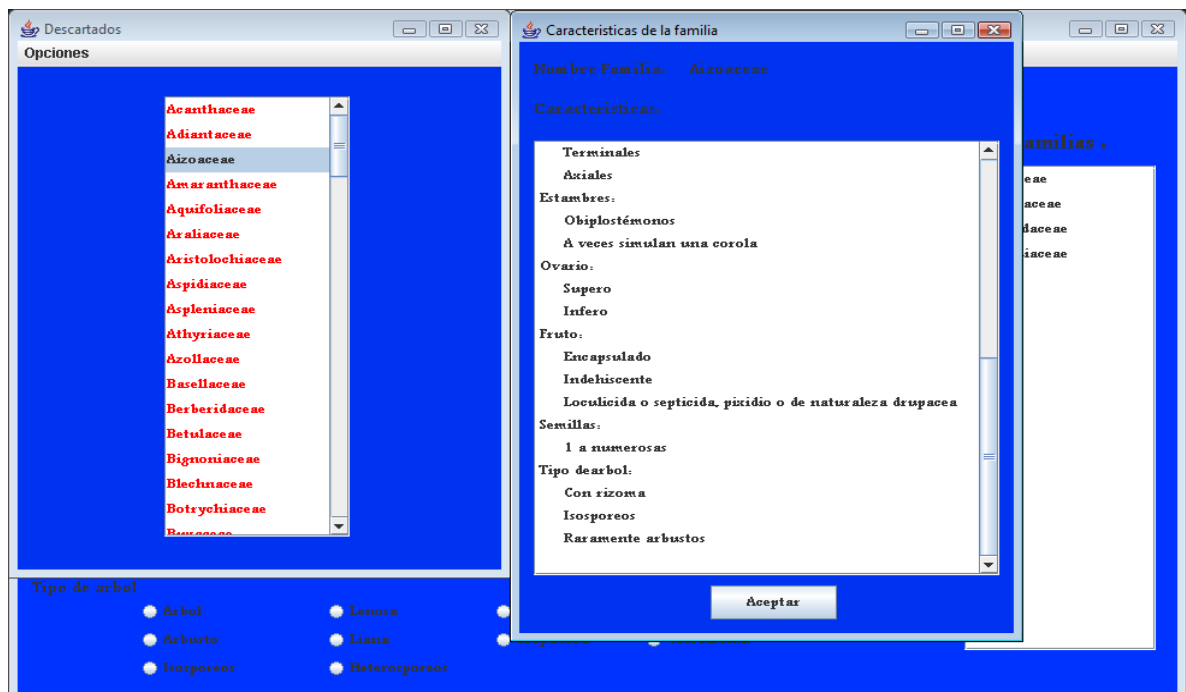
Aquí se pueden ver las características de esta familia seleccionada. Para cerrar esta ventana solo será necesario pulsar el botón Aceptar. Ya que, es muy importante no pulsar el aspa porque sino se cerrara el programa entero.

Si queremos conocer las familias descartadas, solo tendremos que pulsar en el menú Opciones de la pantalla principal y seleccionar la opción de Descartados. Y nos aparecerá una pantalla como la siguiente:



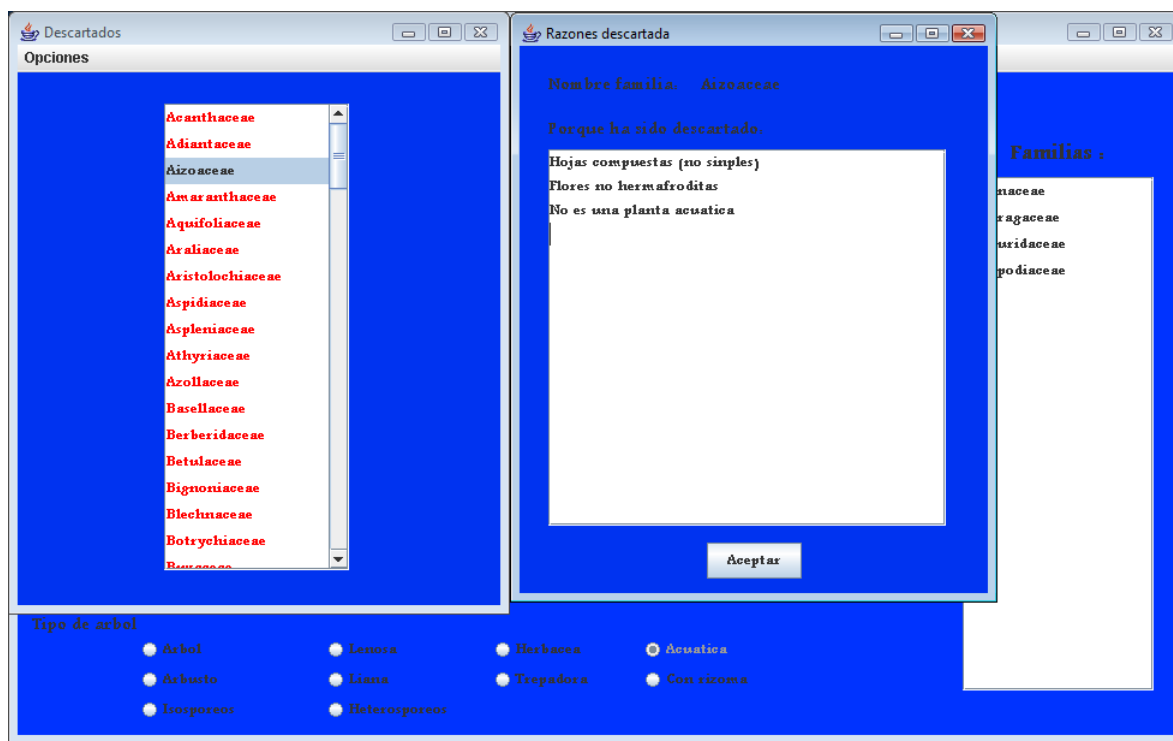
Aquí se muestran en color rojo las familias que han sido descartadas una vez que hemos seleccionado alguno de los botones de la pantalla principal.

Seleccionando alguna de las familias podemos conocer la característica de esta pulsando Opciones→Características.



Para cerrar la ventana que nos ha aparecido es necesario pulsar el botón Aceptar no en el aspa que cerraremos todo el programa.

También podemos conocer las razones por las que esa familia seleccionada ha sido descartada pulsando en Opciones→Razones descartadas.



Para cerrar la ventana que nos ha aparecido es necesario pulsar el botón Aceptar no en el aspa que cerraremos todo el programa.

Para cerrar la ventana con todas las familias descartadas es necesario pulsar Opciones→ Cerrar. Será necesario hacer esto para cerrar la ventana ya que es muy importante no pulsar el aspa ya que el programa se cerrara completamente.

Nosotras pensamos que la herramienta es poco interactiva por lo que debemos de hacerlo más interactiva.

La novedad que hemos introducido ha sido no tener que pulsar en el menú de Opciones -> Comenzar sino que una vez se ejecuta la herramienta aparecen en el cuadro de texto de la derecha todas las familias de plantas.

Entonces ya podemos comenzar a usar la herramienta sin tener que realizar ninguna otra tarea para empezar a usarla.

Lo único que hemos cambiado ha sido eliminar del código de la clase Princip.java el elemento grafico JMenuItem2 que se inicializaba como opción del menú Opciones de la herramienta. Este elemento tenia como texto Comenzar y realizaba el evento de colocar las familias en el cuadro de texto que aparece en la parte derecha de la herramienta.

Para que dichas familias nos aparezcan sin la necesidad de pulsar la opción eliminada en el constructor de Princip.java añadimos la línea this.p.Poner (Familias). Este método de la clase PanelPrin.java se le pasa como parámetro el ArrayList de familias en cada posición contiene un array en el que la primera posición tiene el nombre de la familia y en sucesivas posiciones de dicho array las características de la familia.

El método Poner que esta en la clase PanelPrin.java lo único que realiza es coger el primer elemento de cada array del ArrayList pasado como parámetro y ponerlo en el cuadro de texto que aparece en el cuadro de texto de la parte derecha de la herramienta.



## **NOVEDAD EN LA HERRAMIENTA**

Con este programa conseguimos dar respuesta al usuario según las distintas características que seleccione. No tiene porque responder a todas ellas ya que si desconoce alguna no tiene porque seleccionarla. Además nuestro programa nos proporcionara las familias o familia a la que pertenece la planta que estamos determinando. También podemos conocer las características de las familias que estén en el cuadro de texto, así como las razones por las que otras familias han sido descartadas.

Esta herramienta nos proporciona una manera fácil y sencilla de averiguar la familia a la que pertenece la planta. Ya que tanto la lectura del documento de texto como la búsqueda dicotomica son métodos ineficientes para realizar su determinación.

Como ventaja frente a la búsqueda dicotómica, nuestro programa es el pionero en mostrar la familia a la que pertenece una determinada planta sin tener que responder a todas las preguntas solicitadas para proporcionar una repuesta. Por ello nuestro programa es el pionero en determinar una planta en la familia a la que pertenece teniendo en cuenta que algunas preguntas pueden no tener respuesta.

Como ventaja frente a la lectura del documento es evidente la lectura del documento seria una tarea muy difícil ya que el documento es extenso y además algunas de las características que mencionan resultan desconocidas para personas sin conocimiento en la materia. Con todo esto, este documento muestra el significado de las características que mostramos en la herramienta, además lo que se pretende es introducir esto en la pagina web o como ayuda al usuario en la herramienta con el fin de proporcionarle una ayuda a la hora de responder acerca de características que inicialmente serian desconocidas para el, dándole a conocer la explicación de cada una de ellas.

Ahora realizamos un programa que contemple lo denominado Lógica Borrosa.



# HERRAMIENTA BORROSA

Primeramente comenzaremos dando una visión de la lógica borrosa, ya que antes de realizar esta herramienta necesitamos hacer un estudio de esta disciplina.

Para ello usamos una herramienta que describimos a continuación con la que conseguimos proporcionar un grado de pertenencia a las familias de la flora ibérica. Mediante esta herramienta conseguimos obtener en orden de mayor a menor de grados de pertenencia las familias de la flora ibérica que según sus características se parezcan más a las características seleccionadas por el usuario.

La herramienta usada es Xfuzzy, esta herramienta nos proporciona una interfaz sencilla de usar para poder realizar programas en los que quieras usar lógica borrosa.

La herramienta se puede descargar en la siguiente página:

[http://www.imse.cnm.es/Xfuzzy/index\\_sp.html](http://www.imse.cnm.es/Xfuzzy/index_sp.html)

En ella también se puede descargar un manual de usuario en español para usar la herramienta.

Aunque nosotros proporcionamos a continuación un manual de usuario para usar dicha herramienta en la que explicamos de una manera sencilla el uso de dicha herramienta.

Este manual resulta muy sencillo de leer, y como lo hemos realizado después de haber usado la herramienta sin tener ningún conocimiento previo será de gran ayuda para principiantes.

Ahora explicare los pasos que hemos seguido nosotras en el uso de esta herramienta.

Inicialmente tenemos que hacer un estudio previo de los datos que tenemos que introducir.

Para ello necesitamos conocer el significado de las reglas de inferencia, para ello nos resulta de gran ayuda el conocimiento de nuestro director de proyecto acerca de la lógica borrosa y agradeciéndole la explicación conseguimos entender el significado de las reglas de inferencia.

# LOGICA BORROSA

La lógica borrosa (Fuzzy Logic) ha surgido como una herramienta lucrativa para el control de subsistemas y procesos industriales complejos, así como también para la electrónica de entretenimiento y hogar, sistemas de diagnóstico y otros sistemas expertos. Aunque la lógica borrosa se inventó en Estados Unidos el crecimiento rápido de esta tecnología ha comenzado desde Japón y ahora nuevamente ha alcanzado USA y también Europa. La lógica borrosa es todavía un boom en Japón, el número de cartas patentando aplicaciones aumenta exponencialmente. Principalmente se trata de aplicaciones más bien simples de lógica borrosa.

Lo borroso ha llegado a ser una palabra clave para vender. Los artículos electrónicos sin componentes borrosos se están quedando gradualmente desfasados. Como una mordaza, que muestra la popularidad de la lógica borrosa, cada vez es más frecuente un sello con "fuzzy logic" impreso sobre el producto.

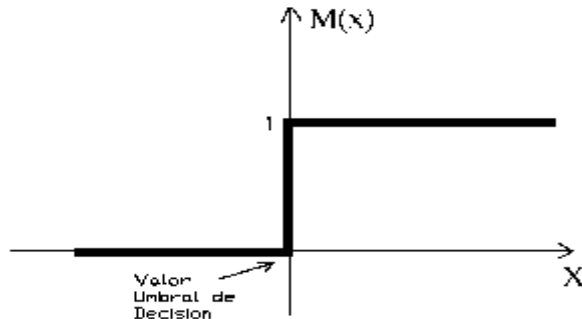
En Japón la investigación sobre lógica borrosa es apoyada ampliamente con un presupuesto enorme. En Europa y USA se están realizando esfuerzos para alcanzar al tremendo éxito japonés. Por ejemplo, la NASA emplea lógica borrosa para el complejo proceso de maniobras de acoplamiento.

La lógica borrosa es básicamente una lógica multievaluada que permite valores intermedios para poder definir evaluaciones convencionales como sí/no, verdadero/falso, negro/blanco, etc. Las nociones como "más bien caliente" o "poco frío" pueden formularse matemáticamente y ser procesados por computadoras. De esta forma se ha realizado un intento de aplicar una forma más humana de pensar en la programación de computadoras. La lógica borrosa se inició en 1965 por Lotfi A. Zadeh, profesor de ciencia de computadoras en la Universidad de California en Berkeley.

Un conjunto de la lógica tradicional se puede expresar mediante un conjunto difuso cuya función de pertenencia sería:

$$\mu_F(x) = \begin{cases} 0 & \text{si } x \in F \\ 1 & \text{si } x \notin F \end{cases}$$

Es decir, una función escalón centrada en el valor umbral de decisión.



Sea F un conjunto difuso definido sobre el universo U.

$$F = \{ (u, \mu_F(u)) \mid u \in U \}$$

que indica que F está formado por todos los pares ordenados "u" y el resultado de la función de pertenencia para todo elemento u dentro del universo de discurso U. La notación que eligió Zadeh para describir los conjuntos difusos es la siguiente:

● si el universo es **discreto**:

$$F = \sum \frac{\mu_F(u)}{u}$$

● si el universo es **continuo**.

$$F = \int \frac{\mu_F(u)}{u}$$

¡Cuidado con esta notación! El sumatorio o la integral **pierden su significado habitual**. En lógica difusa quieren simbolizar una mera enumeración de **tuplas**. Una tupla es un par ordenado como el ya visto (u, MF(u)). La fracción tampoco indica quebrado, simplemente separa los dos elementos de la tupla. Así, el conjunto difuso discreto, "Tirada alta del dado" se define por :

$$F = \{ 0/1 + 0/2 + 0.3/3 + 0.6/4 + 0.9/5 + 1/6 \}$$

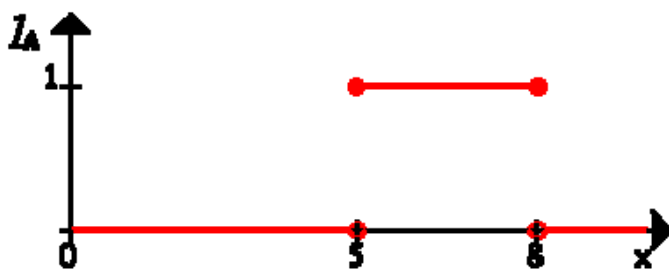
La parte derecha de la tupla indica el elemento y la parte izquierda el grado de pertenencia.

### Un ejemplo:

En primer lugar consideramos un conjunto  $X$  con todos los números reales entre 0 y 10 que nosotros llamamos el universo de discurso. Ahora, definimos un subconjunto  $A$  de  $X$  con todos números reales en el rango entre 5 y 8.

$$A = [5, 8]$$

Ahora mostramos el conjunto  $A$  por su función característica, es decir esta función asigna un número 1 o 0 al elemento en  $X$ , dependiendo de si el elemento está en el subconjunto  $A$  o no. Esto conlleva a la figura siguiente:



Nosotros podemos interpretar los elementos que han asignado el número 1 como los elementos que están en el conjunto  $A$  y los elementos que han asignado el número 0 como los elementos que no están en el conjunto  $A$ .

Este concepto es suficiente para muchas áreas de aplicación. Pero nosotros podemos encontrar fácilmente situaciones donde carece de flexibilidad. Para comprender este concepto veamos un ejemplo:

Queremos describir el conjunto de gente joven. Más formalmente nosotros podemos denotar

$$B = \{\text{conjunto de gente joven}\}$$

Como - en general - la edad comienza en 0, el rango más inferior de este conjunto está claro. El rango superior, por otra parte, es más bien complicado de definir. Como un primer intento colocamos el rango superior en, digamos, 20 años. Por lo tanto nosotros definimos B como un intervalo denominado:

$$B = [0,20]$$

Ahora la pregunta es: ¿ por qué alguien es en su 20 cumpleaños joven y al día siguiente no? Obviamente, este es un problema estructural, porque si movemos el límite superior del rango desde 20 a un punto arbitrario podemos plantear la misma pregunta.

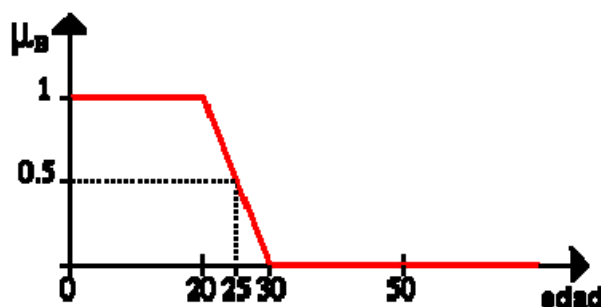
Una manera más natural de construir el conjunto B estaría en suavizar la separación estricta entre el joven y el no joven. Nosotros haremos esto para permitir no solamente la (crispada) decisión "él/ella SI está en el conjunto de gente joven" o "él/ella NO está en el conjunto de gente joven", sino también las frases más flexibles como "él/ella SI pertenece un poquito más al conjunto de gente joven" o "él/ella NO pertenece aproximadamente al conjunto de gente joven".

Pasamos a continuación a mostrar como un conjunto borroso nos permite definir una noción como "él/ella es un poco joven".

Tal y como constatamos en la introducción podemos usar conjuntos borrosos para hacer computadoras más sabias, y ahora tenemos que codificar la idea más formalmente. En nuestro ejemplo primero codificamos todos los elementos del Universo de Discurso con 0 o 1. Una manera de generalizar este concepto está en permitir más valores entre 0 y 1. De hecho, nosotros permitimos infinitas alternativas entre 0 y 1, denominando el intervalo de unidad  $Y_0 = [0, 1]$ .

La interpretación de los números ahora asignados a todos los elementos del Universo de Discurso es algo más difícil. Por supuesto, el número 1 asignado a un elemento significa que el elemento está en el conjunto B y 0 significa que el elemento no está definitivamente en el conjunto el B. El resto de valores significan una pertenencia gradual al conjunto B.

Para ser más concretos mostramos ahora gráficamente el conjunto de gente joven de forma similar a nuestro primer ejemplo por su función característica.



De esta forma unos 25 años de edad todavía sería joven al grado de 50 por ciento.

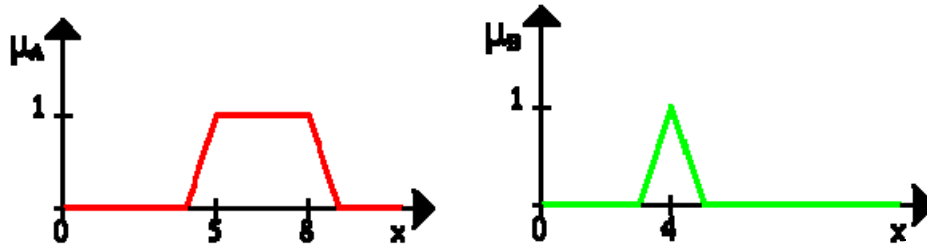
### Operaciones con conjuntos borrosos

Ahora que tenemos una idea de lo que son conjuntos borrosos, podemos introducir las operaciones básicas sobre conjuntos borrosos. Parecido a las operaciones sobre conjuntos booleanos nosotros también podemos interseccionar, unificar y negar conjuntos borrosos. En su primerísimo artículo sobre conjuntos borrosos, L. A. Zadeh sugirió el operador mínimo para la intersección y el operador máximo para la unión de dos conjuntos borrosos. Es fácil ver que estos operadores coinciden con la unificación booleana, e intersección si nosotros únicamente consideramos los grados miembros 0 y 1.

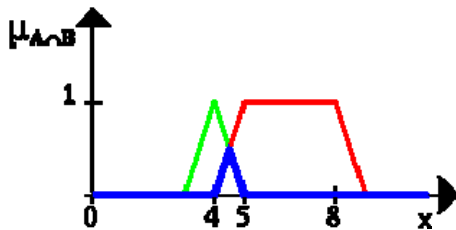
En lógica difusa hay muchas maneras de definir estas operaciones. Cualquier operación que cumpla las restricciones de una **T-Norma** puede ser usada para intersectar, igual que cualquier **S-Norma** puede ser usada para unir conjuntos difusos. Las T-Normas especifican un conjunto de condiciones que deben reunir aquellas operaciones que deseen ser usadas para intersectar conjuntos, mientras que las S-Normas hacen lo propio para las uniones.

Las operaciones lógicas más usuales son la **unión**, la **intersección** y el **complemento**.

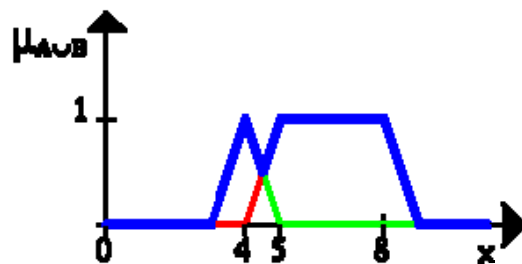
A fin de aclarar esto, mostraremos varios ejemplos. Sea A un intervalo borroso entre 5 y 8, y B un número borroso entorno a 4. Las figuras correspondientes se muestran a continuación:



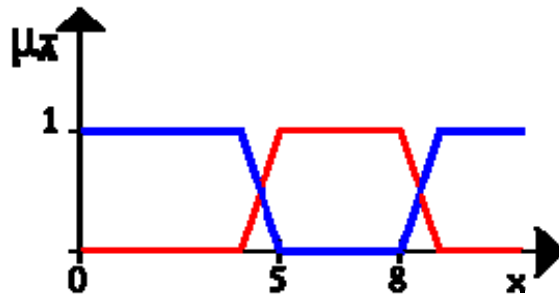
La figura siguiente muestra la operación AND (Y) del conjunto borroso A y el número borroso B (el resultado es la línea azul).



La operación OR (O) del conjunto borroso A con el número borroso B se muestra en la próxima figura (nuevamente, es la línea azul).

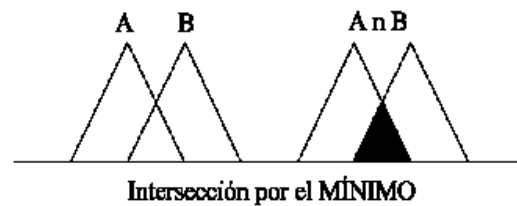


Esta figura da un ejemplo para una negación. La línea azul es la NEGACION del conjunto borroso A.



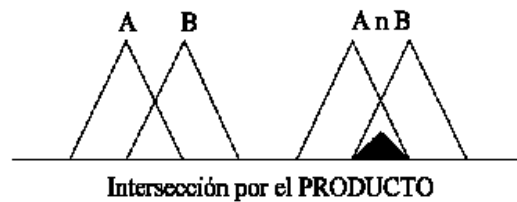
En la práctica, se suele usar siempre como T-Norma el mínimo o el producto de las dos funciones de pertenencia, y como S-Norma el máximo de ambas o su suma, porque computacionalmente el coste es mucho menor. Sin embargo hay ocasiones donde puede hacer falta una T-Norma más complicada. Sobre esta cuestión me refiero a [1], donde además se detalla un criterio para escoger entre las dos posibilidades que ofrecemos a continuación para cada operación.

Algebraicamente, las dos formas de expresar la intersección difusa se expresan:



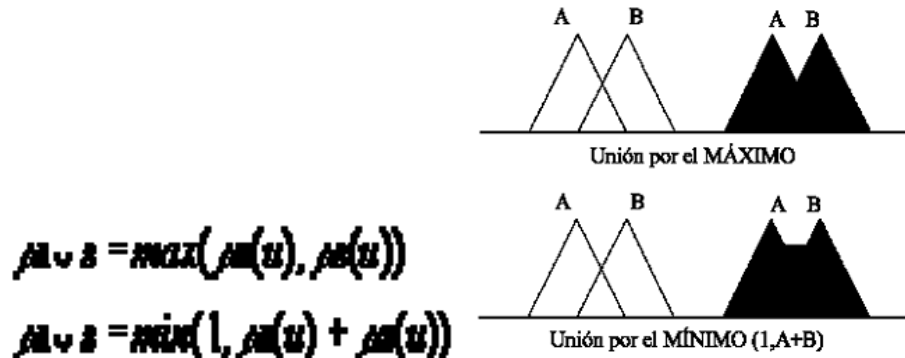
$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(x))$$

$$\mu_{A \cap B} = \mu_A(x) \cdot \mu_B(x)$$





Análogamente, para la unión tenemos:



y para el complemento,

$$\mu_{\bar{A}} = 1 - \mu_A(u)$$

Ahora podemos estudiar las diferentes familias de conectivas lógicas.

Una familia de conectivos lógicos borrosos (T, S, N) está formada por una norma triangular T, una conorma triangular S y una negación N, y se denomina una terna de De Morgan o terna de Morgan cuando S es la t-conorma dual de T respecto a la negación N.

Las familias de conectivos lógicos mas utilizadas son:

	$T(x, y)$	Generador aditivo	$S(x, y)$	Generador aditivo	$N(x)$
<u>Zadeh</u>	$\text{Min}(x, y)$		$\text{Max}(x, y)$		$1-x$
<u>Yager</u> <sub><math>\lambda</math></sub>	$1 - \text{Min}(((1-x)^\lambda + (1-y)^\lambda)^{1/\lambda}, 1)$	$-\log(x)$	$\text{Min}((x^\lambda + y^\lambda)^{1/\lambda}, 1)$	$-\log(1-x)$	$(1-x^\lambda)^{1/\lambda}$
<u>Dombi</u> <sub><math>\lambda &gt; 0</math></sub>	$\frac{1}{1 + \left[ \left( \frac{x}{1-x} \right)^\lambda + \left( \frac{y}{1-y} \right)^\lambda \right]^{1/\lambda}}$	$\left[ \frac{1-x}{x} \right]^\lambda$	$\frac{1}{1 + \left[ \left( \frac{x}{1-x} \right)^\lambda + \left( \frac{y}{1-y} \right)^\lambda \right]^{1/\lambda}}$	$\left[ \frac{x}{1-x} \right]^\lambda$	$1-x$
<u>Weber</u> <sub><math>\lambda &gt; -1</math></sub>	$\text{Max}\left(\frac{x+y-1+\lambda xy}{1+\lambda}, 0\right)$	$1 - \frac{\ln(1+\lambda x)}{\ln(1+\lambda)}$	$\text{Min}\left(\frac{x+y+\lambda xy}{1+\lambda}, 1\right)$	$\frac{\ln(1+\lambda x)}{\ln(1+\lambda)}$	$\frac{1-x}{1+\lambda x}$
<u>Weber</u> <sub><math>\lambda &lt; -1</math></sub>	$\text{Max}\left(\frac{x+y-1+\lambda xy}{1+\lambda}, 0\right)$	$1 - \frac{\ln(1+\lambda x)}{\ln(1+\lambda)}$	$\text{Min}\left(\frac{x+y+\lambda xy}{1+\lambda}, 1\right)$	$\frac{\ln(1+\lambda x)}{\ln(1+\lambda)}$	$\frac{1-x}{1+\lambda x}$
<u>Łukasiewicz</u> <u>Weber</u> <sub>con <math>\lambda=0</math></sub>	$\text{Max}(x+y-1, 0)$		$\text{Min}(1, x+y)$		$1-x$
<u>Hamacher</u> <sub><math>\gamma &gt; 0</math></sub>	$\frac{xy}{\gamma + (1-\gamma)(x+y-xy)}$	$\frac{1}{\gamma} \ln \frac{\gamma + (1-\gamma)x}{x}$	$\frac{x+y-xy-(1-\gamma)xy}{1-(1-\gamma)xy}$	$\frac{1}{\gamma} \ln \frac{\gamma + (1-\gamma)(1-x)}{1-x}$	$1-x$

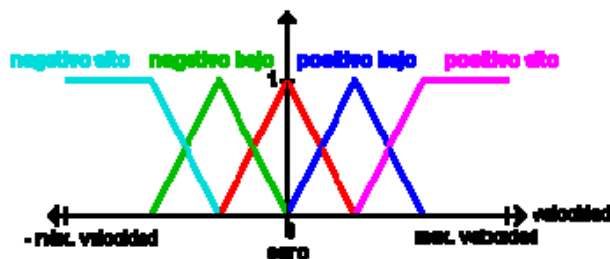
## El control borroso

Los controladores borrosos son las aplicaciones más importantes de la teoría borrosa. Ellos trabajan de una forma bastante diferente a los controladores convencionales; el conocimiento experto se usa en vez de ecuaciones diferenciales para describir un sistema. Este conocimiento puede expresarse de una manera muy natural, empleando las variables lingüísticas que son descritas mediante conjuntos borrosos.

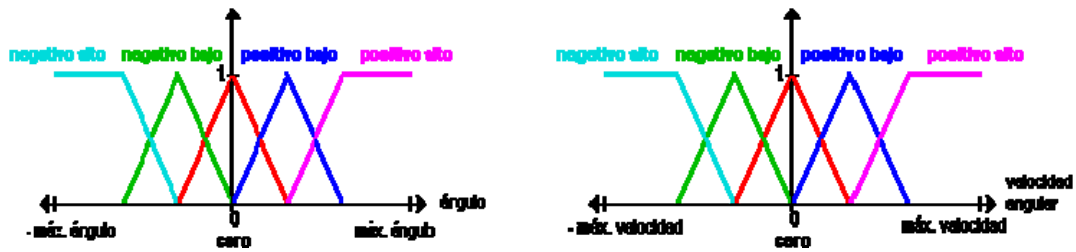
### Ejemplo: El péndulo invertido

El problema está en equilibrar una pértiga sobre una plataforma móvil que puede moverse en dos únicas direcciones, a la izquierda o a la derecha. Ante todo, nosotros tenemos que definir (subjetivamente) cual es la velocidad del anden: alta, baja, etc. Esto se hace para especificar las funciones pertenecientes al conjunto borroso:

- negativo alto (celeste)
- negativo bajo (verde)
- cero (rojo)
- positivo bajo (azul)
- positivo alto (morado)



Lo mismo se hace para el ángulo entre la plataforma y la pértiga, además de para la velocidad angular de este ángulo:



Apréciase que, para hacerlo más fácil, suponemos que al principio la pértiga está en una posición cercana a la central para que un ángulo mayor de, digamos, 45 grados en cualquier dirección no pueda - por definición - ocurrir.

Ahora daremos varias reglas que dicen qué hacer en situaciones concretas:

Considere por ejemplo que la pértiga está en la posición central (el ángulo es cero) y no se mueve (la velocidad angular es cero). Obviamente esta es la situación deseada, y por lo tanto no tenemos que hacer nada (la velocidad es cero).

Consideremos otro caso: el polo está en la posición central como antes, pero está en movimiento a baja velocidad en la dirección positiva. Naturalmente nosotros tendríamos que compensar el movimiento de la pértiga moviendo la plataforma en la misma dirección a baja velocidad.

De esta forma hemos constituido dos reglas que pueden ponerse en una forma más formalizada como esta:

**Si** el ángulo es cero **y** la velocidad angular es cero **entonces** la velocidad será cero.  
**Si** el ángulo es cero **y** la velocidad angular es positiva baja **entonces** la velocidad será positiva baja.

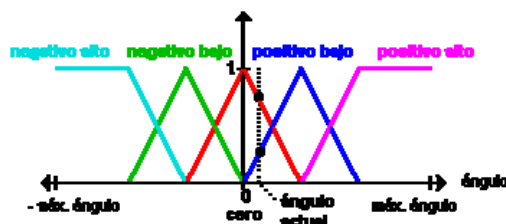
Podemos resumir todas las reglas aplicables en una tabla:

		ángulo				
veloc		NA	NB	C	PB	PA
áng	NA			NA		
	NB			NB	C	
	C	NA	NB	C	PB	PA
	PB		C	PB		
PA		PA				

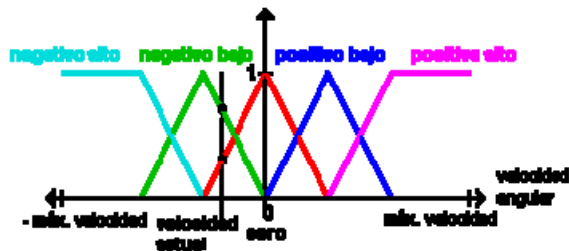
donde NA es una (usual) abreviatura para negativa alta, NB para negativa baja, etc.

A continuación mostraremos como estas reglas pueden aplicarse con valores concretos para el ángulo y velocidad angular. Para ello vamos a definir dos valores explícitos para el ángulo y la velocidad angular para operar con ellos. Consideremos la situación siguiente:

Un valor actual para el ángulo:



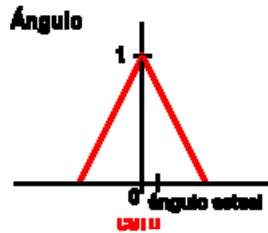
Un valor actual para la velocidad angular:



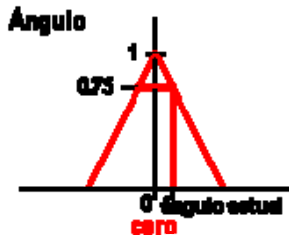
Ahora mostraremos como aplicar nuestras reglas a esta situación real. Veamos como aplicar la regla

**Si** el ángulo es cero **y** la velocidad angular es cero **entonces** la velocidad será cero.

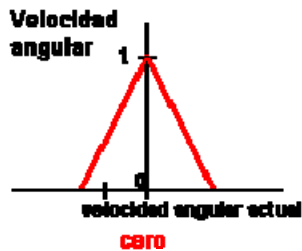
a los valores que hemos definido. Esta es la variable lingüística "ángulo" donde nos centramos en el conjunto "cero" y el ángulo actual:



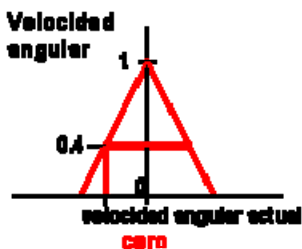
Nos damos cuenta que nuestro valor real pertenece al conjunto borroso "cero" en un grado de 0.75:



Ahora mostramos la variable lingüística "velocidad angular" donde nos centramos en el conjunto borroso "cero" y el valor actual de velocidad angular:

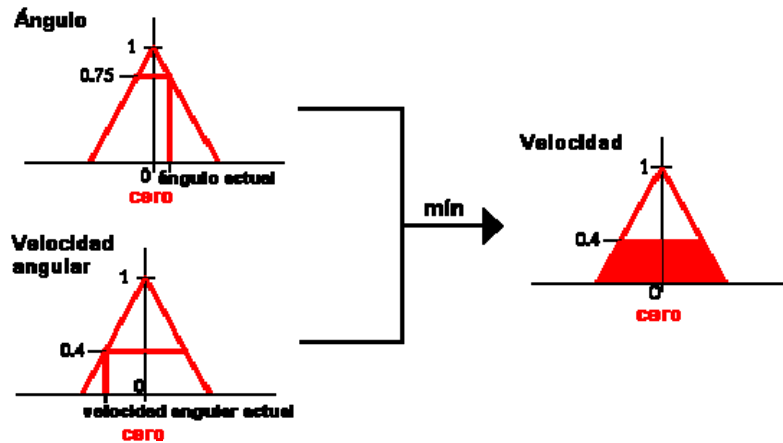


Nos damos cuenta que nuestro valor real pertenece al conjunto borroso "cero" en un grado de 0.4:

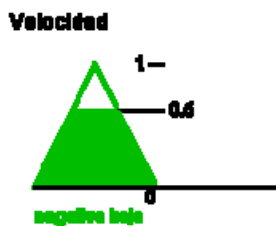


Como las dos partes de la condición de nuestra regla están unidas por una Y (operación

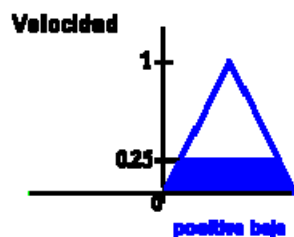
lógica AND) calculamos el  $\min(0.75, 0.4) = 0.4$  y cortamos el conjunto borroso "cero" de la variable "velocidad" a este nivel (según nuestra regla):



Por su parte, el resultado de la regla **Si** el ángulo es cero **y** la velocidad angular es negativa **baja entonces** la velocidad será negativa **baja** es:



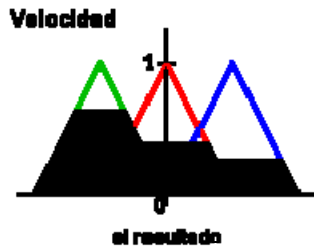
El resultado de la regla **Si** el ángulo es cero **y** la velocidad angular es positiva **baja entonces** la velocidad será positiva **baja** es:



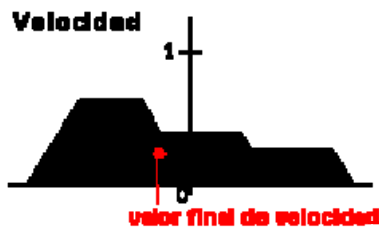
El resultado de la regla  
**Si** el ángulo es positivo bajo **y** la velocidad angular es negativa baja **entonces** la velocidad será  
 es: cero



Estas cuatro reglas solapadas desembocan en un resultado único:



El resultado del controlador borroso es un conjunto borroso (de velocidad), así que tenemos que escoger un valor representativo como salida final. Hay varios métodos heurísticos (métodos de claridad o defuzzification), uno de ellos es tomar el centro de gravedad del conjunto borroso:



El procedimiento completo se denomina controlador de Mamdani.

Aplicaciones de la lógica borrosa

Principalmente, miraremos la aptitud del control borroso en términos generales.

El empleo del control borroso es recomendable:

- Para procesos muy complejos, cuando no hay un modelo matemático simple.
- Para procesos altamente no lineales.
- Si el procesamiento del (lingüísticamente formulado) conocimiento experto puede ser desempeñado.

El empleo del control borroso no es una buena idea si:

- El control convencional teóricamente rinde un resultado satisfactorio.
- Existe un modelo matemático fácilmente soluble y adecuado.
- El problema no es soluble.

Definiciones

### **Intersección de Conjuntos**

Nosotros llamamos un nuevo conjunto generado desde dos conjuntos determinados A y B, intersección de A y B, si el nuevo conjunto contiene exactamente esos elementos que están contenidos en A y en B.

### **Unificación de Conjuntos**

Nosotros llamamos un nuevo conjunto generado desde dos conjuntos determinados A y B, unificación de A y B, si el nuevo conjunto contiene todos los elementos que se contienen en A o en B o en ambos.

### **Negación de Conjuntos**

Nosotros denominamos al nuevo conjunto que conteniendo todos los elementos que están en el universo de discurso pero no en el conjunto A la negación de A.



### **Variables lingüísticas**

Un variable lingüística es un quintuple  $(X, T(X), U, G, M)$ , donde  $X$  es el nombre de la variable,  $T(X)$  es el término conjunto (es decir, el conjunto de nombres de valores lingüísticos de  $X$ ),  $U$  es el universo de discurso,  $G$  es la gramática para generar los nombres y  $M$  es un conjunto de reglas semánticas para asociar cada  $X$  con su significado.

# HERRAMIENTA XFUZZY

Después nos descargamos de la pagina web mencionada anteriormente el archivo .jar para poder instalarlo en nuestro ordenador. Una vez hecho esto tuvimos que leernos el manual de usuario que también se puede descargar de la pagina web en formato pdf (Acrobat Reader) y que esta escrito en español, de gran ayuda ya que normalmente los manuales de las herramientas técnicas están escritos en ingles.

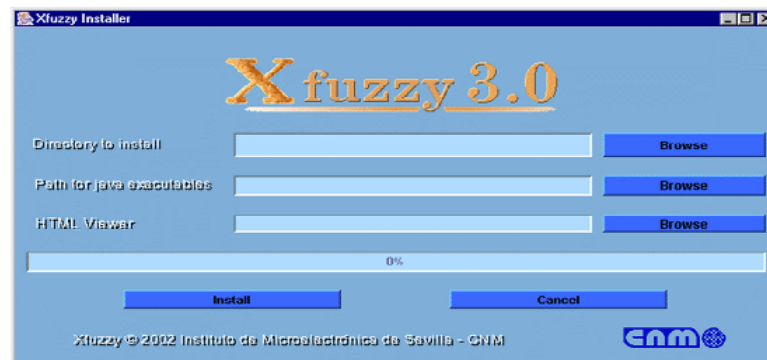
Mostramos una imagen con la guía de instalación de la herramienta:

## Requisitos del sistema:

Xfuzzy 3.0 puede ser ejecutado sobre cualquier plataforma que disponga del "Java Runtime Environment" (JRE). Para definir nuevos paquetes de funciones es también necesario disponer de un compilador Java. La última versión del "Java Software Development Kit", incluyendo el JRE, un compilador Java y otras herramientas relacionadas, puede encontrarse en <http://java.sun.com/j2se/>.

## Guía de Instalación:

- Descargue el fichero [XfuzzyInstall.jar](#).
- Ejecute este fichero. Sobre MS-Windows basta con pulsar sobre el icono. En general el fichero se ejecuta con el comando `"java -jar XfuzzyInstall.jar"`. Esto abrirá la siguiente ventana de instalación.



- Elija un directorio para instalar Xfuzzy. Si el directorio no existe se creará en el proceso de instalación.
- Elija el directorio de los ejecutables de java (java, javac, jar, etc.). Este directorio suele ser el subdirectorio "/bin" de la instalación del J2SE.
- Elija un navegador para mostrar los ficheros de ayuda.
- Pulse en el botón "Install" para descomprimir la distribución de Xfuzzy en el directorio base seleccionado.
- Los ejecutables de Xfuzzy residen en el directorio "/bin".
- Los ficheros ejecutables son ficheros de comandos. Si cambia la localización de la distribución de Xfuzzy deberá repetir el proceso de instalación.

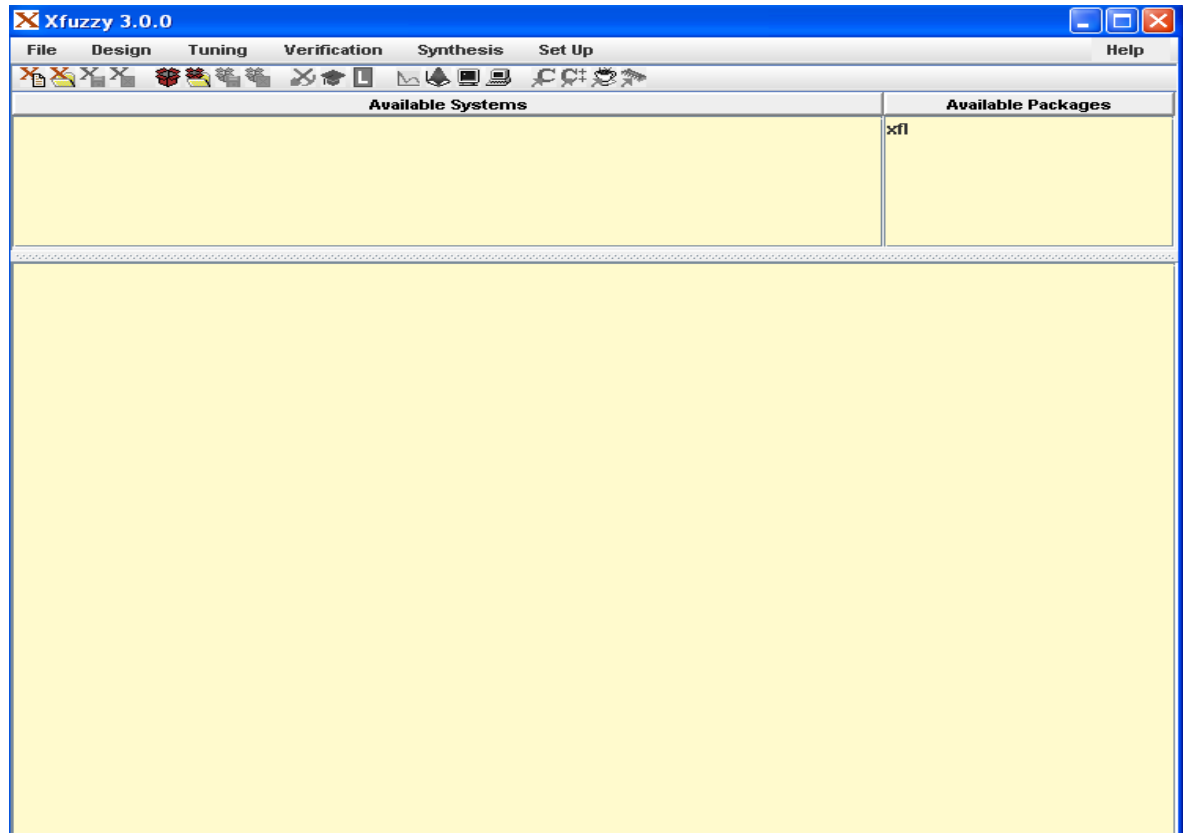
Nos descargamos la última versión de la herramienta Xfuzzy 3.0 cuyo lenguaje de especificación XFL3 es un lenguaje expresivo y que realiza más tareas que la anterior versión, como por ejemplo la definición de nuevos operadores difusos por parte del usuario.

También proporciona vistas en 2D y 3D para la visualización de las salidas.

Además y lo mas importante ha sido programado en Java así que puede ser instalado en cualquier ordenador que tenga instalado el Java Runtime Environment (JRE). Podemos descargarnos una última versión de esto también en dicha página web.

Para comenzar a usar la herramienta debemos de hacer doble clic en el archivo que esta en la carpeta bin del directorio donde hayamos instalado el Xfuzzy y pulsar en el archivo xfuzzy, siendo este un archivo de tipo MS-DOS.

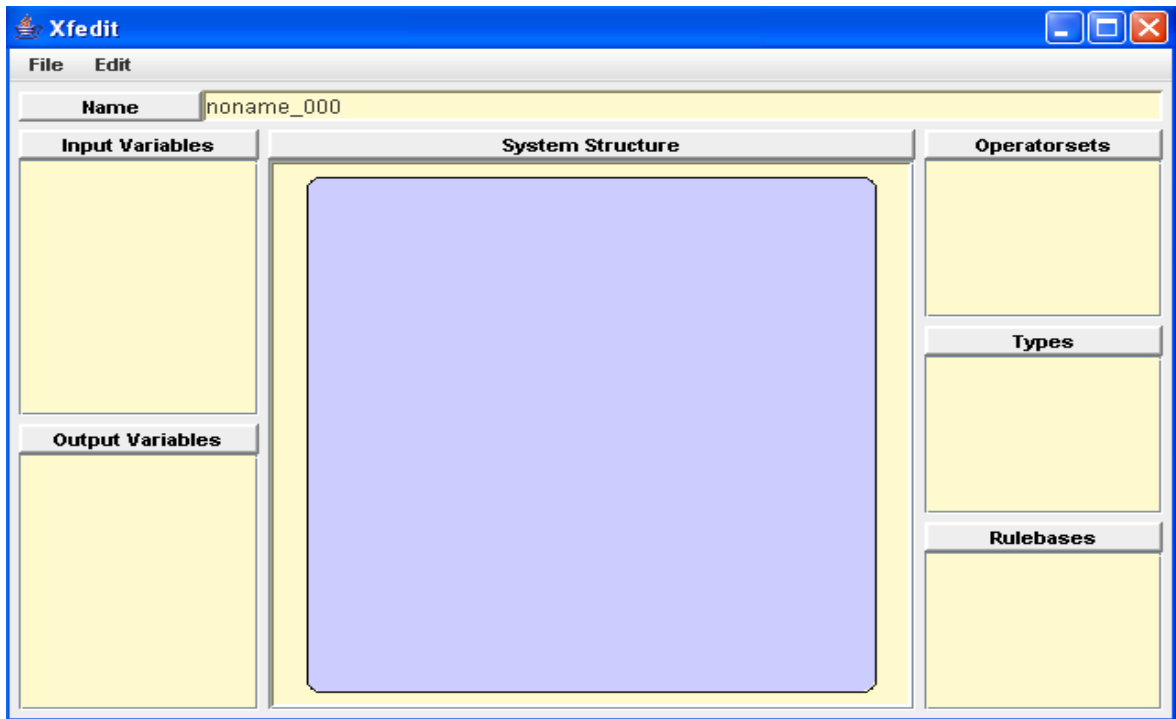
Una vez pulsado esto nos saldrá una pantalla como la siguiente:



Entonces tendremos que pulsar en el menú File y dar en la opción de New System.

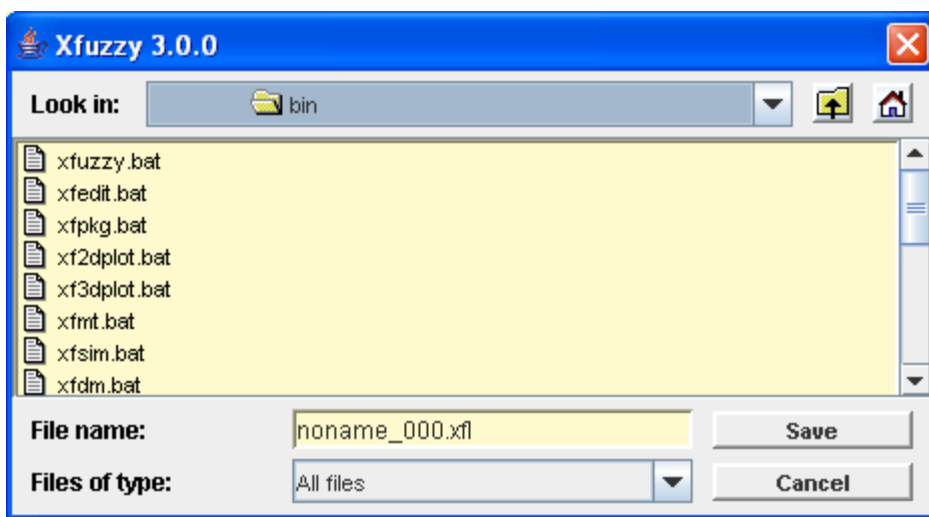
Cuando pulsas en esto te aparece como nombre de sistema noname\_000 que para cambiarlo tenemos que irnos al menú Design y escoger la opción Edit System aunque previamente debemos de haber seleccionado el sistema a editar.

Nos aparecerá una ventana como esta:



A continuación para cambiar el nombre del sistema debemos de pulsar en la opción File SaveAs y entonces será cuando te aparezca una ventana como esta; introduces el nombre que desees pero siempre con la extensión .xfl.

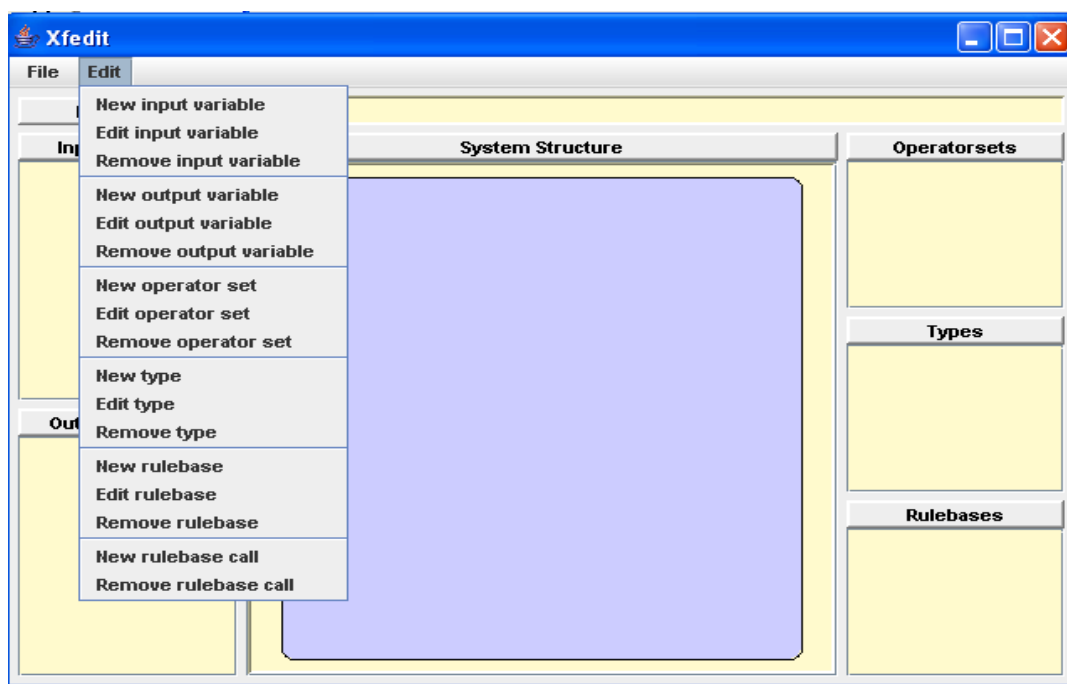
En nuestro caso ponemos proyectoSI.xfl



En la parte superior vemos como te aparece el nombre del sistema que tú le has proporcionado, en nuestro caso el sistema se llama proyectoSI.

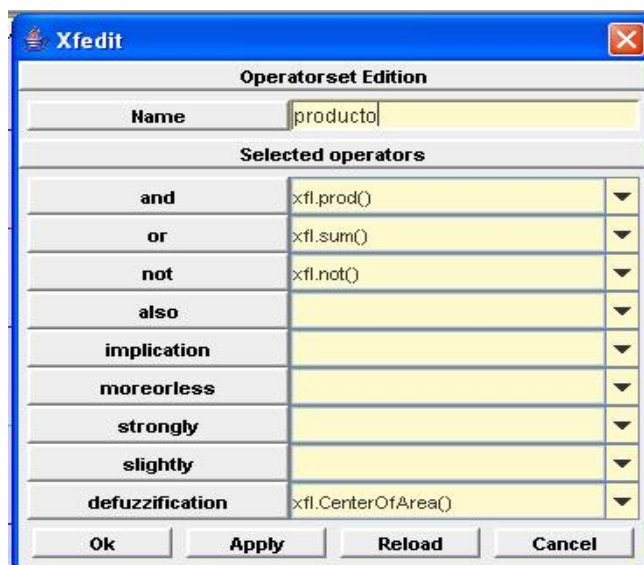
Lo primero que debemos de realizar en este sistema es la introducción de los operadores. Para ello debemos de pulsar en el menú de Edit y elegir la opción de New operator set.

En esta imagen vemos todas las opciones del menú Edit.

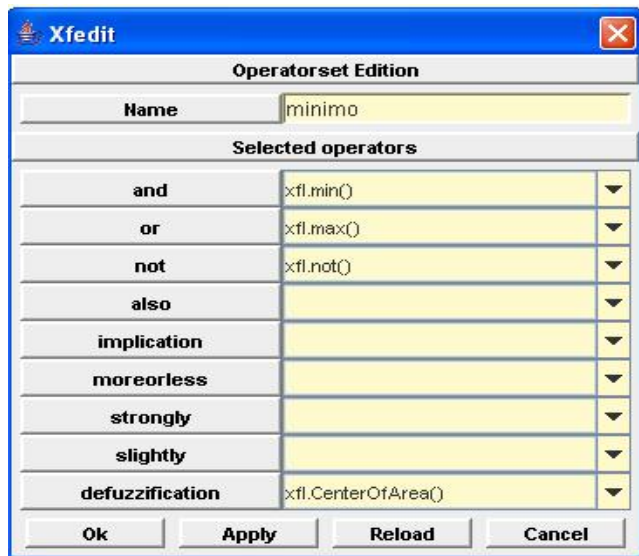


Nuestros operadores van a ser las Reglas de inferencia explicadas anteriormente, tanto el producto, como el mínimo como la Regla de inferencia de lukasiewicz.

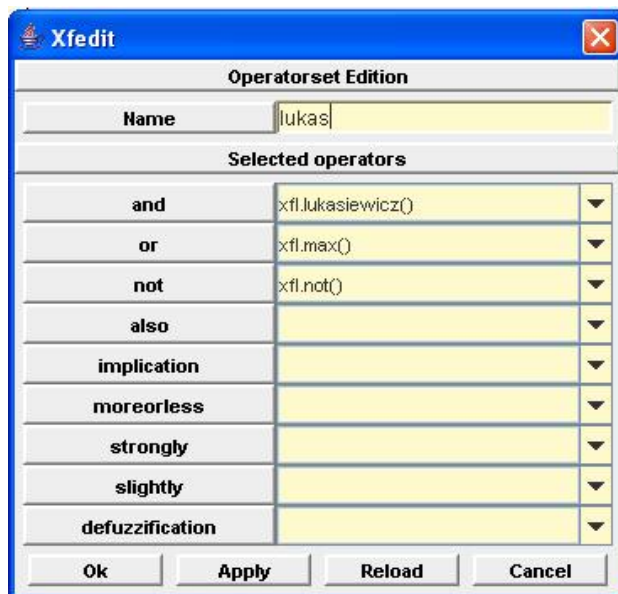
Para conocer como realizamos este paso nosotras, solo debe de mirar la siguiente imagen:



Aquí realizamos un operador para el producto.



En esta imagen se puede apreciar el operador para el mínimo.



Y por ultimo en esta imagen se aprecia el último operador el de Lukasiewicz.

A continuación volvemos a la pantalla principal. El siguiente paso a realizar será comenzar creando los tipos de lógica borrosa a utilizar, para ello se pulsa en el menú Edit y elegimos la opción de New Type.

En nuestro caso necesitábamos 4 tipos.

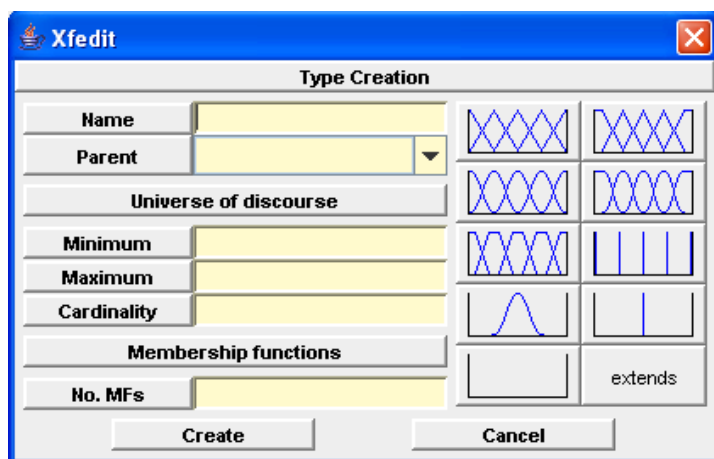
Uno de los tipos, tBool, es para todas las características booleanas que tenemos en nuestra herramienta como Caduca, Perenne, etc... Para ellas solo puede existir un tipo booleano o se cumple la característica o no se cumple, es cierta o es falsa no hay mas posibilidad para dichas características.

También debemos de tener otro tipo, tPlanta, para cada una de las familias de la flora ibérica. Ya que cada planta debe de tener este tipo concreto. Por lo tanto a este tipo van a pertenecer 116 plantas.

Además debemos de tener otros dos tipos, Testambres y Tpetalos, para añadir a las características del programa inicial dos características borrosas.

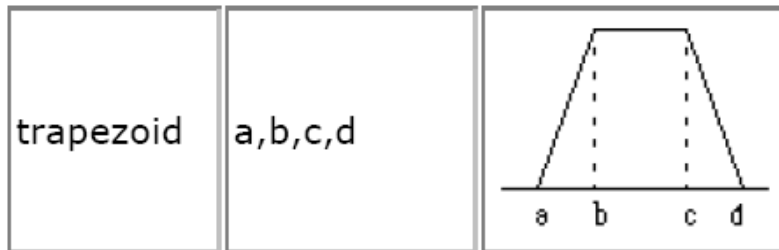
Nosotros estamos construyendo en este momento un sistema totalmente borroso con lo que la anterior herramienta nos resulta de gran ayuda. Para conseguir una manera heterogénea de seguir usando las características por las que hemos discretizado las familias anteriormente debemos de poder seguir utilizándolas. Por ello añadimos un tipo booleano para cada una de las características de la herramienta hecha inicialmente. Ya que para proporcionar un grado de pertenencia para cada una de las familias y seguir usando estas características debemos añadir todas las características como datos de entrada.

Entonces cuando pulsamos en el menú Edit la opción New Type nos aparece una ventana como esta.



En ella elegimos el nombre que le vamos a dar al tipo en nuestro caso tBool. Después señalamos la pestaña Parent o directamente como va a ser un Trapezoid señalamos en la pantalla de la derecha el que corresponde a este tipo de función.

Y a continuación explicaremos los parámetros para la función trapezoidal que con la siguiente imagen que le mostramos será de fácil comprensión.

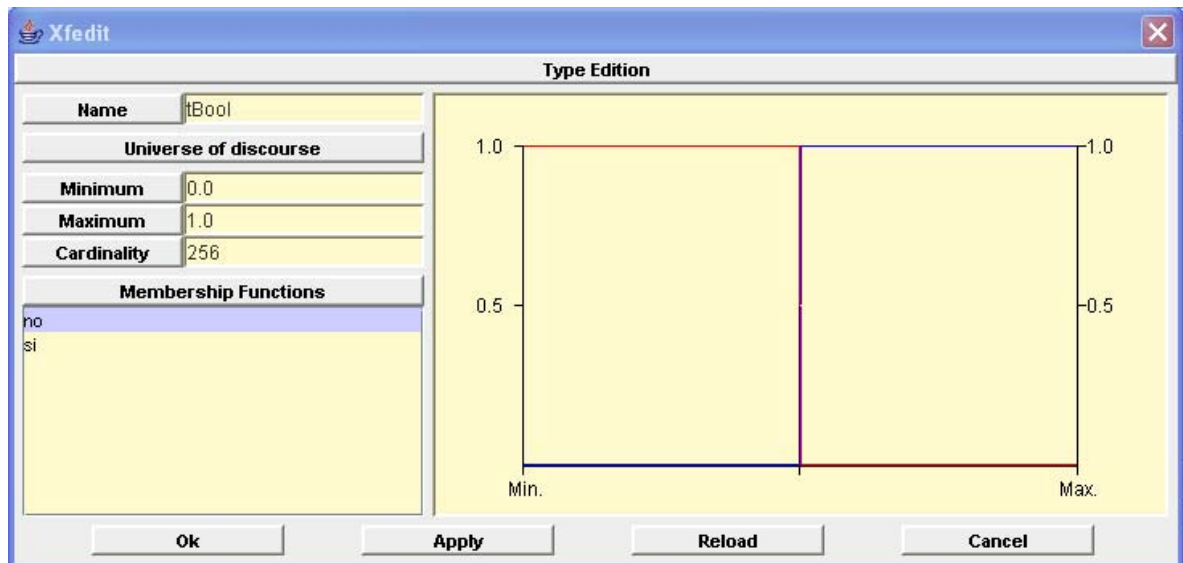


El parámetro a sirve para señalar el comienzo del trapecio en el eje de coordenadas. El siguiente parámetro, b sirve para señalar el fin de la subida izquierda del trapecio. El parámetro, c para conocer el punto desde el que el trapecio empieza a descender, parte derecha de este. Y el parámetro d para señalar el fin del trapecio.

A continuación debajo del título que pone Universo of discourse ponemos el mínimo a 0.0 y el máximo a 1.0, además la cardinalidad se deja por defecto.

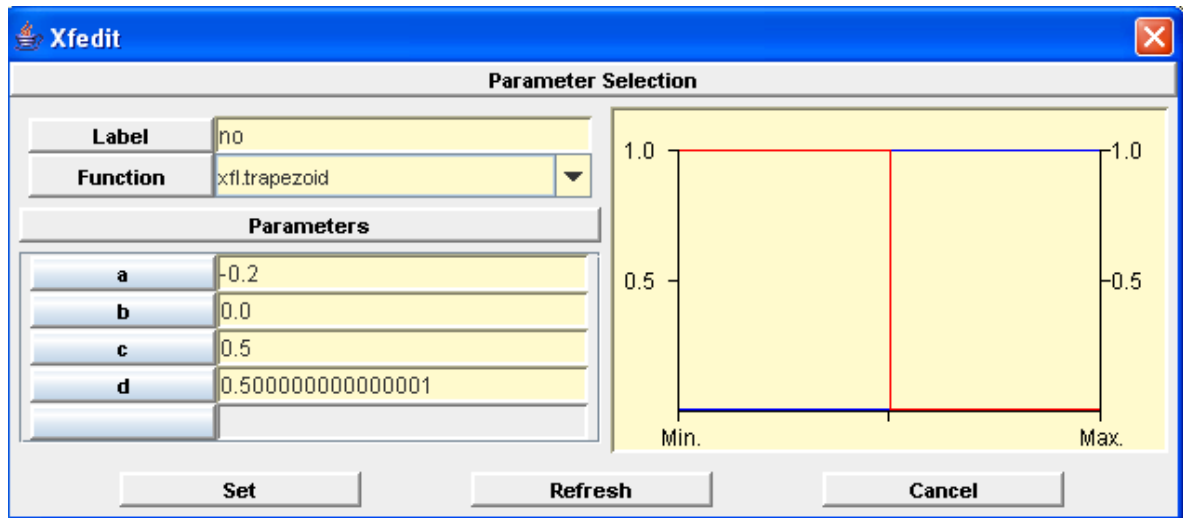
Añadimos dos Membership Functions en nuestro caso dos y le damos al botón Create.

Una vez hecho esto nos aparece una ventana como esta.





Entonces para dar valor a los Membership Function debemos de pulsar en cada uno de ellos, un doble clic ya que inicialmente te salen nombres por defecto y viendo la siguiente pantalla:



Cuando señalas en alguna de ellas la línea que la representa se ve de color rojo para conocer los valores que toma.

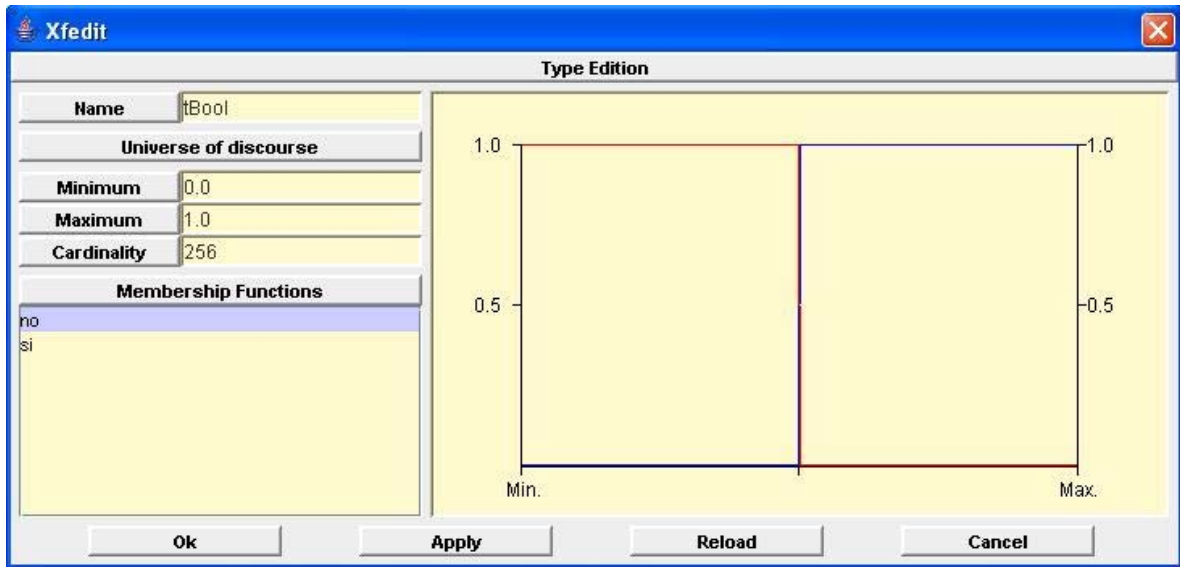
Entonces cambiamos el nombre de la función en la pestaña Label y además debemos de cambiar los valores de a, b, c y d para que nos quede una función como esta, estos valores son los que anteriormente hemos descrito.

Después de cambiar los valores damos al botón Refresh para ver la modificación de los cambios y después pulsamos el botón Set.

Esto lo debemos de realizar para cada uno de los Membership Function en nuestro caso ahora solo con dos.

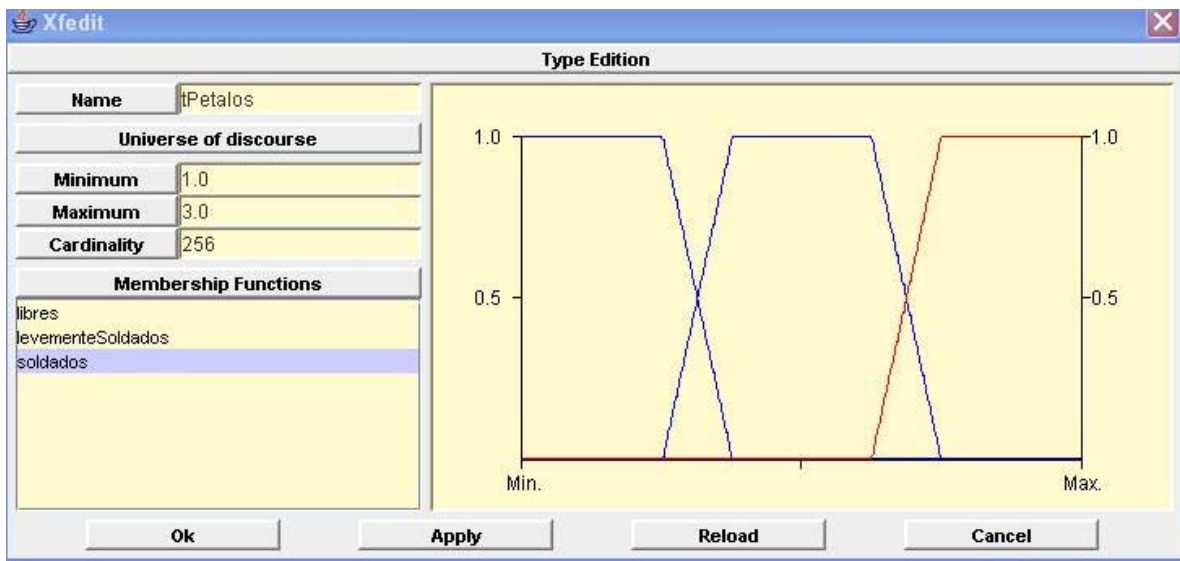
Una función es para cuando se cumple la variable booleana, es decir es true y otra para cuando la variable booleana no se cumple, es decir false.

Una vez hecho esto con las dos funciones nos quedara una pantalla como esta.



Para volver a la pantalla principal debemos de pulsar en el botón de Ok.

El siguiente tipo es el tPetalos:

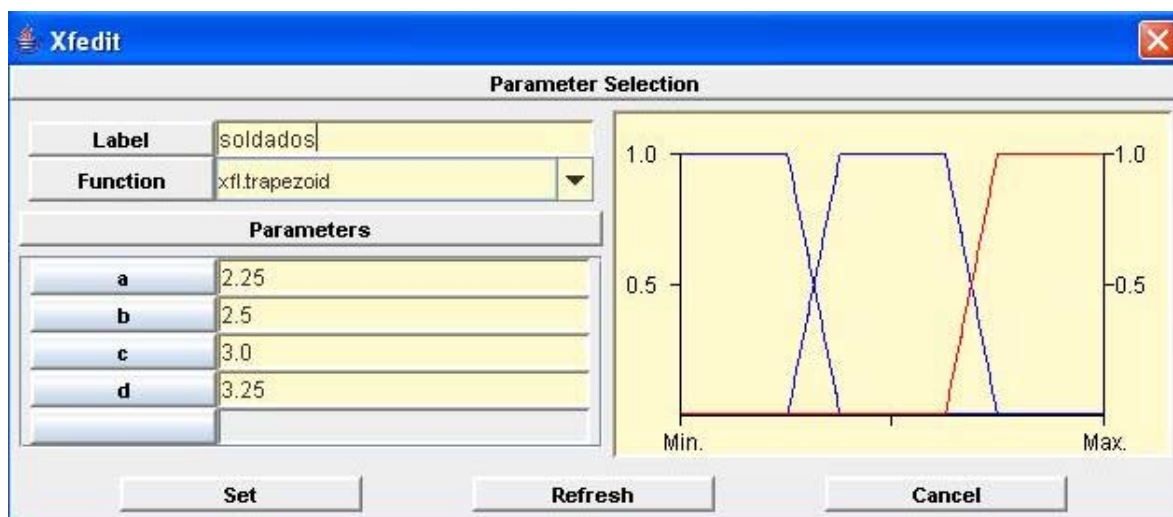


El nombre dado al tipo es Tpetalos. El mínimo de este tipo es 1 y el máximo es 3. La cardinalidad se la dejamos por defecto.

Podemos observar que en el apartado Membership Functions aparecen tres funciones. Una para cada una de las descripciones que puede tener la característica Pétalos. Estos pueden estar libres, levemente soldados o soldados.

Para cada una de estas descripciones se tiene que crear una función. Como podemos observar la señalada en este caso es soldados y muestra la figura en su parte derecha y pintada de color rojo la grafica que le corresponde a esta función con sus distintos valores.

Para ello anteriormente a esto debemos de introducir los parámetros de esta función.

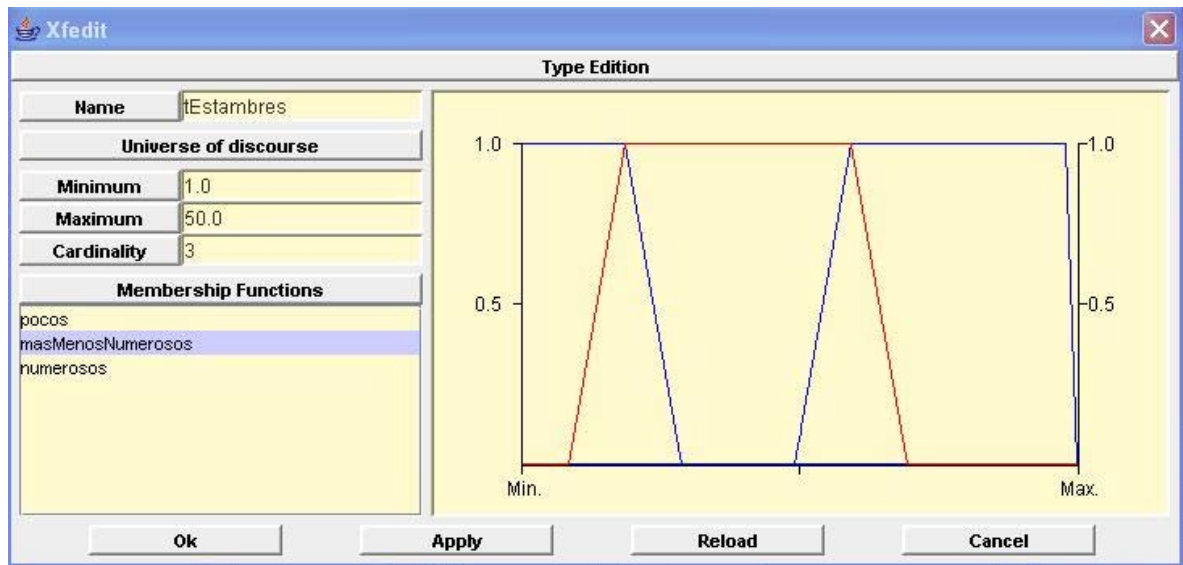


Inicialmente le proporcionamos un nombre, en este caso soldados.

Pero como se puede observar el trapezio no tiene parte derecha descendiente esto es debido a que el parámetro c le hemos introducido un valor superior 3.25 al máximo introducido y explicado anteriormente de la función total que es 3 por lo tanto el trapezio se quedara incompleto.

Esto también ocurre con la parte izquierda del trapezio que resulta para la función libres que como se puede observar en la figura inicial. Ya que la característica libres debe de tener un valor inferior a 1 en el parámetro a o incluso para el parámetro b. Lo que si conocemos observando la figura es que el parámetro c es mayor que 1 porque se muestra en dicha figura.

Para el otro tipo, Testambres debemos de realizar lo mismo. Esta figura muestra lo que hemos hecho:

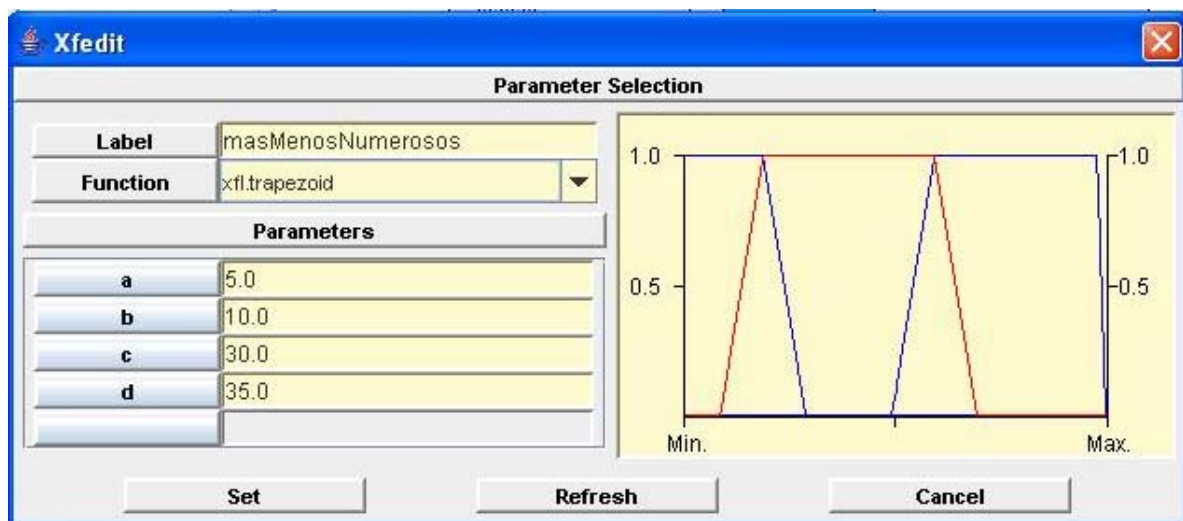


Podemos observar que el nombre de este tipo es tEstambres. La t es por tipo y Estambres porque el tipo va a pertenecer a la característica de estambres.

El mínimo de esta función es 1 y el máximo 50 debido a que según la tabla de Excel que generamos al principio del desarrollo de este proyecto observamos que estos son los valores entre los que se encuentran el número de estambres.

La cardinalidad en este caso la asignamos el numero 3.

Las tres Membership Functions son pocos, masMenosNumerosos o numerosos. También debemos de editar cada una de estas funciones y vamos a mostrar una de ellas, masMenosNumerosos:



Como vemos el nombre coincide con el nombre de la función.

La forma de la función con la que vamos a representarla es con la trapezoidal. Y los parámetros son los descritos en la figura.

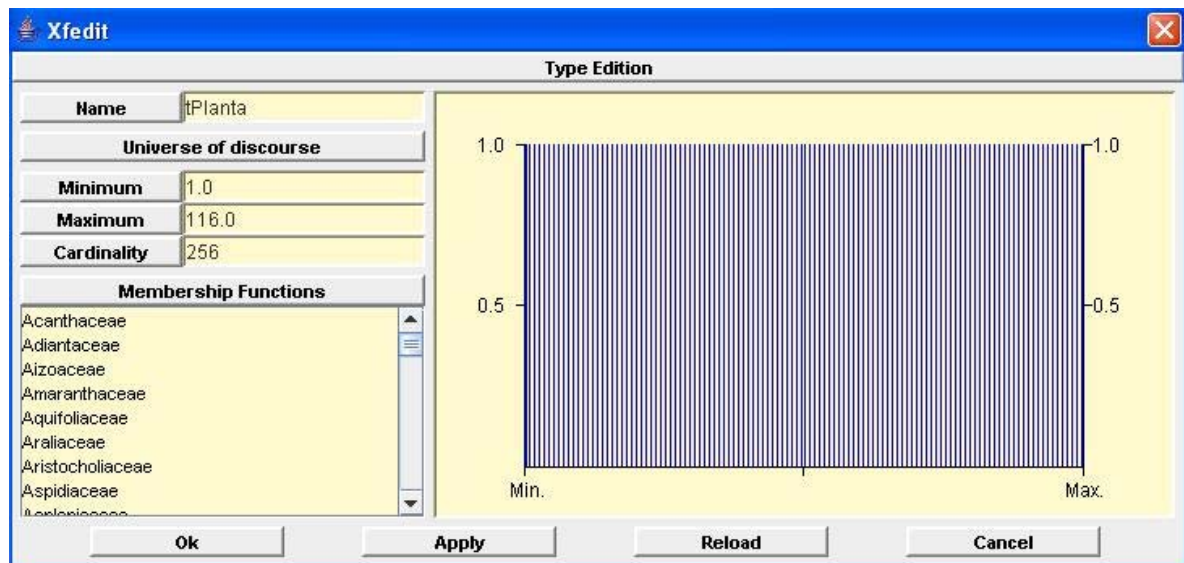
También observamos que la señalada con color rojo es la que corresponde a esta función, que posee los parámetros que le hemos introducido.

Esto se debe de realizar para cada una de las funciones que aparezcan en las Membership Functions, para poder generar así la función global, que la observamos en la parte derecha de la imagen anterior en la que se muestran las tres graficas para las tres funciones.

Ahora para el tipo planta debemos de realizar otro tipo diferente llamado tPlanta, este tipo elegimos una función denominada Singleton que solo posee un parámetro a.

Para este tipo debemos de tener una Membership Function para cada una de las familias con lo que resultan ser 116 funciones una para cada una de las familias.

Para cada una de ellas el parámetro de a ira aumentando, para la *Acanthaceae* el parámetro a vale 1.0 para la *Adiantaceae* vale 2.0 y así hasta las 116 plantas.



Podemos observar que el mínimo de la función es 1 y el máximo de esta es 116 con lo que reunimos el total del numero de familias a cubrir.

La cardinalidad la dejamos por defecto.

Como se puede observar para cada familia tenemos una función.

Esto será el siguiente paso, añadir los datos de entrada, por lo tanto tenemos que introducir un tipo para cada una de las características del programa inicial para poder usar lógica borrosa con ellas también. Además introduciremos dos datos de entrada más, el tipo de pétalos y el número de estambres.

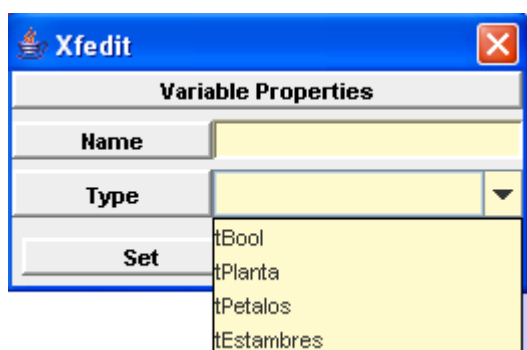
Con lo que resulta que tenemos que introducir como datos de entrada 36 variables de tipo booleano una para cada una de las características, como por ejemplo: inCaduca, para la característica Caduca de la herramienta e in al ser dato de entrada.

También debemos de añadir otros dos datos de entrada, uno con el tipo de pétalos de tipo tPetalos: inPetalos. Y otro con el número de estambres de tipo tEstambres: inEstambres.

Estos datos los tenemos que introducir en la parte izquierda de la pantalla principal bajo el título Input Variables. En ella introduciremos las 36 variables booleanas cada una con el nombre precedido de la palabra in, de input (entrada) y el nombre que tenían en la herramienta inicial. Como por ejemplo: Caduca resulta ser inCaduca.

Para ello debemos de pulsar en el menú Edit y escoger la opción New input variable.

Te aparece una ventana como esta, en la que tienes que poner el nombre de la variable de entrada y un tipo a elegir entre los que hemos realizado:



Después pulsamos el botón Set y así para todas.

Para cada una de estas variables de entrada debemos de darle un tipo en este caso tBool al ser variables de tipo booleano.

Para la entrada inEstambres, le damos el tipo que la corresponde, tEstambres.

Para la entrada inPetalos le damos el tipo tPetalos.

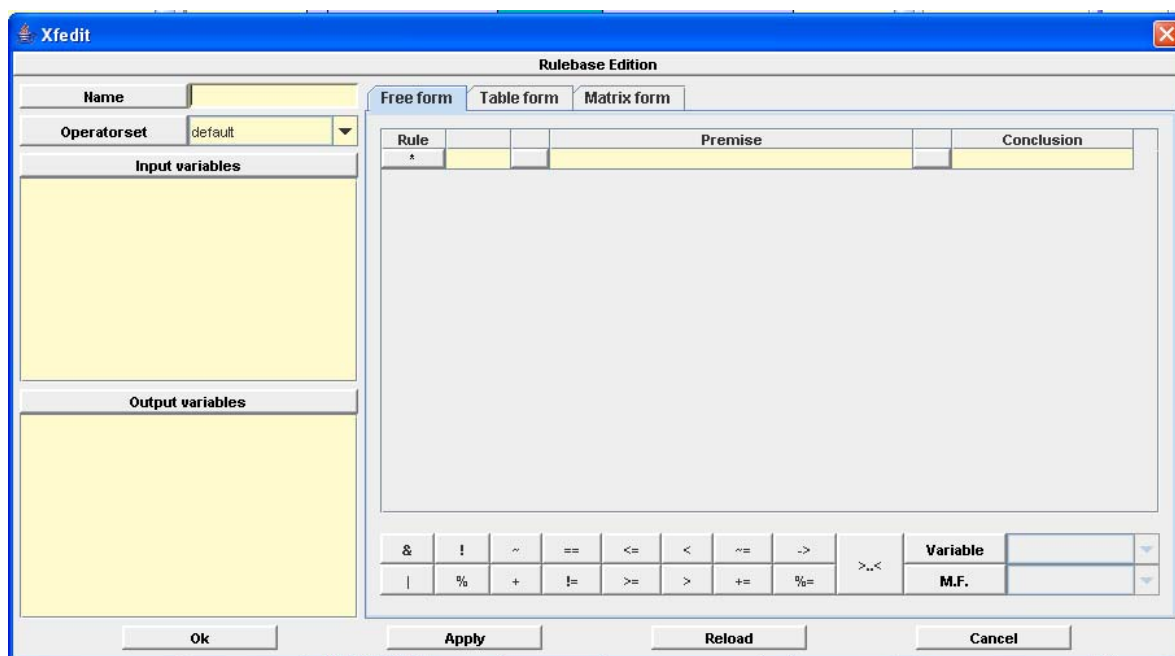
Con ello finalizamos la entrada de variables de entrada.

Ahora introduciremos las variables de salida, que será pinchando en el menú de Edit y eligiendo la opción New output variable. Te aparecerá una ventana como la anterior y en nuestro caso el único tipo de salida será tPlanta y pondremos 3 variables de salida una para cada regla de inferencia, una para el producto, otra para el mínimo y otra para el método de Lukasewicz.

En ella introducimos los tres tipos de salida uno para cada Regla de inferencia con los siguientes nombres: para la del mínimo outPlantaMin, para el producto outPlantaProd y para Lukasewicz, outPlantaLukas.

Ahora realizamos las reglas para obtener el grado de pertenencia a cada familia según las variables de entrada que el usuario haya introducido. Esto se puede ver pulsando en el menú de opciones Edit y eligiendo la que pone New rulebase.

Entonces te aparece una pantalla como esta:



Tuvimos un problema a la hora de mostrarnos las variables de entrada y las de salida en dicha ventana pero en el apéndice problema Xfuzzy lo comentaremos.

A continuación mostramos nuestras reglas para nuestra rulebase rFamiliasProd.



Rule			Premise	Conclusion
0	1.0	if	(inPerenne == si & inOpuesta == si & inEstipulas == no ...	outPlantaProd = Acanthac...
1	1.0	if	(inPerenne == si & inEsteroidal == si & inIsosporico == si )	outPlantaProd = Adiantaceae
2	1.0	if	(inCaduca == si & inPerenne == si & inOpuesta == si & i...	outPlantaProd = Aizoaceae
3	1.0	if	(inPerenne == si & inOpuesta == si & inAlterna == si & i...	outPlantaProd = Amaranth...
4	1.0	if	(inPerenne == si & inAlterna == si & inEstipulas == si & i...	outPlantaProd = Aquifoliac...
5	1.0	if	(inCaduca == si & inPerenne == si & inAlterna == si & in...	outPlantaProd = Araliaceae
6	1.0	if	(inCaduca == si & inPerenne == si & inAlterna == si & in...	outPlantaProd = Aristocholi...
7	1.0	if	(inPerenne == si & inEsteroidal == si & inHerbacea == si...	outPlantaProd = Aspidiaceae
8	1.0	if	(inPerenne == si & inEsteroidal == si & inHerbacea == si...	outPlantaProd = Aspleniace...
9	1.0	if	(inPerenne == si & inEsteroidal == si & inHerbacea == si...	outPlantaProd = Athyriaceae
10	1.0	if	(inCaduca == si & inPerenne == si & inEsteroidal == si &...	outPlantaProd = Azollaceae
11	1.0	if	(inPerenne == si & inAlterna == si & inEntera == si & inH...	outPlantaProd = Basellaceae
12	1.0	if	(inCaduca == si & inPerenne == si & inAlterna == si & in...	outPlantaProd = Berberida...
13	1.0	if	(inCaduca == si & inAlterna == si & inEstipulas == si & i...	outPlantaProd = Betulaceae
14	1.0	if	(inCaduca == si & inPerenne == si & inEstipulas == no &...	outPlantaProd = Bignoniac...
15	1.0	if	(inPerenne == si & inEsteroidal == si & inRizoma == si &...	outPlantaProd = Blechnace...
16	1.0	if	(inCaduca == si & inPerenne == si & inDividida == si & in...	outPlantaProd = Botrychiac...
17	1.0	if	(inCaduca == si & inPerenne == si & inOpuesta == si & i...	outPlantaProd = Burserac...

Le introducimos un nombre en este caso rFamiliasProd y seleccionamos el Operatorset a usar en este caso el producto.

Nos da a elegir entre tres maneras de observar e introducir las reglas.

Para introducirlas es mejor la vista en tabla y para verlas es mejor la vista Free form. Ya que en la vista Table form en cada casilla podemos introducir el valor de las variables entrada. Un valor para cada una de las entradas al sistema, es decir de las 38 (36 booleanas y dos de tipos diferentes).

También podemos ver como la implicación resulta dar un valor para la variable de salida, outPlantaProd.

Por ello tenemos que introducir reglas para las 116 plantas con 38 comprobaciones cada una de ellas.

Como podemos ver en la parte izquierda de la pantalla aparecen las variables de entrada y la variable de salida que corresponde con el operador producto.

Además en la parte derecha aparecen las 116 reglas que hemos realizado.

Aparece 1.0 ya que si se cumplen todas las comprobaciones que hacemos en la regla que ponemos a continuación la variable de salida va a tener un grado de pertenencia a dicha regla de 1.0. Este es el numero mas grande que se puede obtener como valor de grado de pertenencia.



Para realizar estas reglas hemos ido mirando en el fichero plantas.txt, cada vez que se cumple una de las características se pulsa el nombre de la variable de entrada correspondiente y se compara con el valor del tipo que le corresponda. Para las 36 variables booleanas se puede comparar con los valores que posee el tipo tBool si o no. También se le pueden dar alguno de los otros dos tipos de variables, tEstambres y tPetalos con sus correspondientes valores de estas variables. Como por ejemplo: MasoMenosNumerosos en el caso del tipo tEstambres.

Cuando la variable de entrada no esta en la descripción de la familia esta variable simplemente no se pone en la regla.

Cada comparación se une con todas las siguientes comparaciones de la regla con el operador and.

Para finalizar la regla damos una implicación que lo que significa es asignar el nombre de la familia a la que corresponde la regla a la variable de salida outPlantaProd.

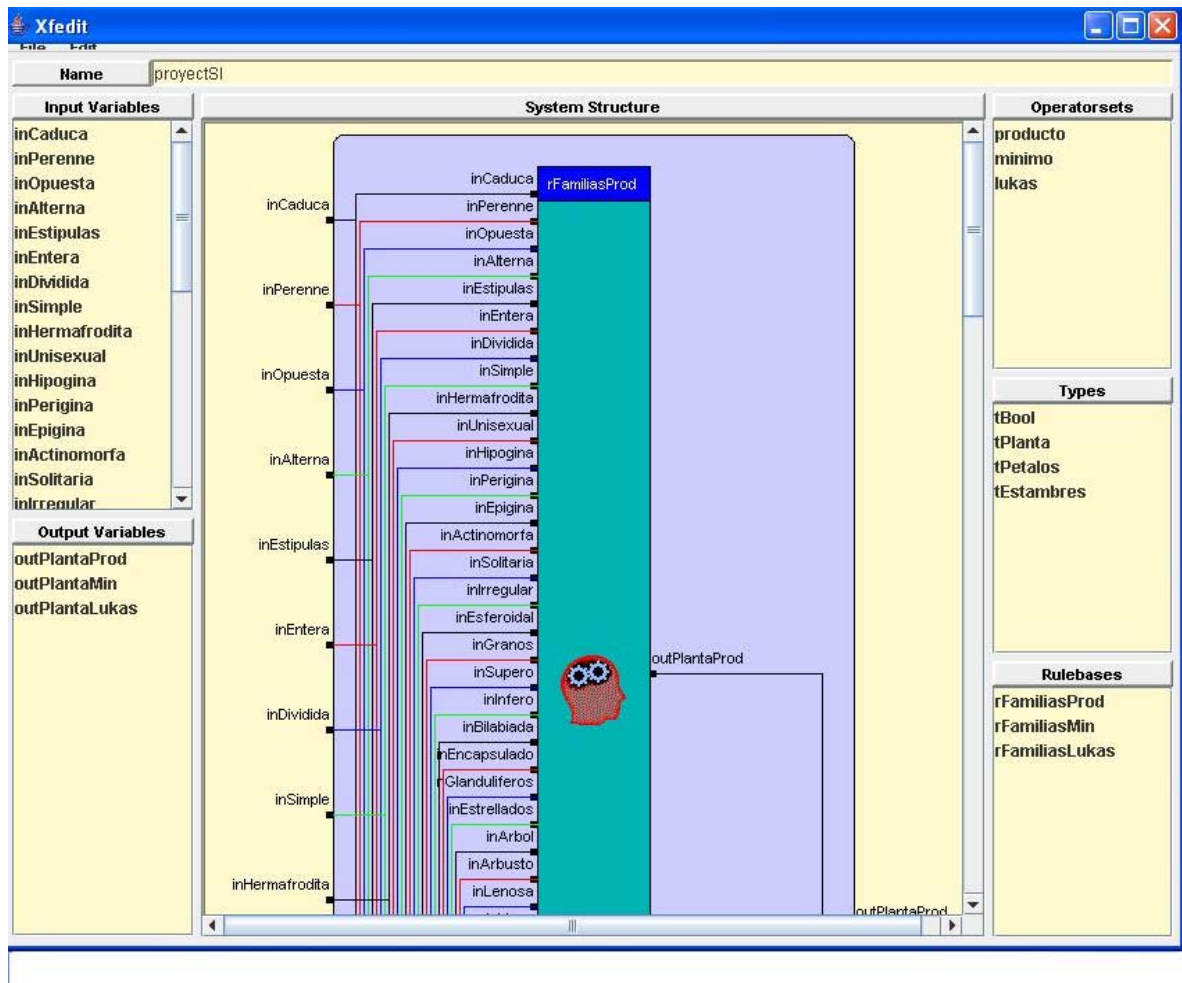
Este proceso se realiza para las 116 familias de las que queremos conocer el grado de pertenencia. Además se realiza para cada una de las tres reglas de inferencia en este caso hemos explicado la del producto pero para el mínimo y para el método lukasewicz lo único que debemos de cambiar es en el espacio que pone OperatorSet cambiarlo por el correspondiente que estemos realizando.

También debemos de cambiar la variable de salida en su caso la que corresponda.

Como he mencionado anteriormente esto lo tenemos que realizar para cada una de las 3 reglas de inferencia, lo que no resulto una tarea sencilla y en un primer momento resulta ser muy difícil pero una vez empezamos con dicha tarea fue resultando fácil mecánicamente.

Para la regla de inferencia mínimo le dimos como nombre a la Rulebase rFamiliasMin y como variable de salida outPlantaMin, para la regla de inferencia de lukasewicz le dimos como nombre a la Rulebase rFamiliasLukas y como variable de salida outPlantaLukas.

Para finalizar mostramos la pantalla principal:



Aquí se puede observar las entradas al sistema en este caso una caja con rFamiliasProd que lo que realiza es según los datos de entrada al sistema da un resultado de salida de la aplicación de las reglas a la variable de salida del sistema outPlantaProd.

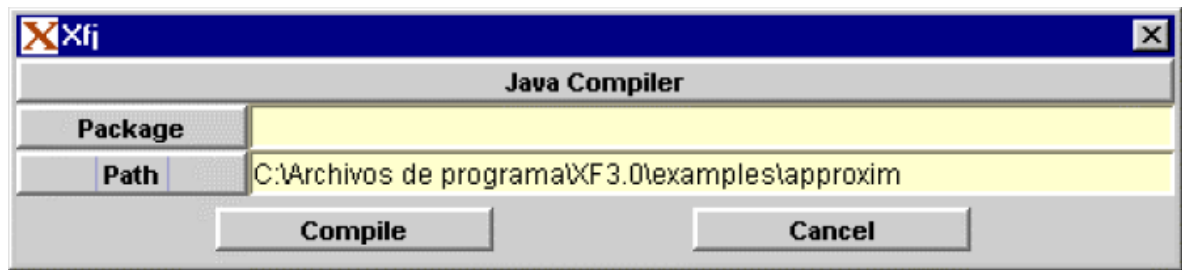
Pero moviendo el cursor se pueden observar todas las Rulebase del sistema.

Dando a la pestaña Edit del menú de opción y eliges la opción de New rulebase call podemos añadir Rulebase a mostrar. Con la opción Remove rulebase call la caja verde con el muqueñito que este señalada se elimina de esta vista.

Después de realizar todo esto, queríamos conocer la manera de poder realizar una interfaz grafica en lenguaje de programación java para que el usuario pudiese usar lógica borrosa, sin tener conocimiento de esta.

En el manual de usuario que proporciona un apartado en el que explica la forma de obtener todo lo que hemos realizado en la herramienta en lenguaje java.

Para ello pulsamos en el menú de opciones Synthesis de la pantalla principal. Y pulsamos la opción de To Java y entonces nos aparece una ventana como esta:



Y añadimos los datos que nos pide generándonos cuatro ficheros: FuzzyInferenceEngine.java, MembershipFunction.java, FuzzySingleton.java y proyectoSI.java.

Este último fichero es diferente en todos los sistemas ya que es debido al nombre que le hemos dado a nuestro sistema.

Palabras textuales del manual:

Los tres primeros ficheros corresponden a descripciones de dos interfaces y una clase que son comunes a todos los sistemas de inferencia difusos.

El último fichero contiene la descripción específica del sistema difuso systemname.xfl.

El fichero FuzzyInferenceEngine.java describe una interfaz Java que define un sistema de inferencia difuso general. Esta interfaz define cuatro métodos para implementar el proceso de inferencia con valores crisp y difusos.

El fichero MembershipFunction.java contiene la descripción de una interfaz usada para describir un número difuso. Contiene sólo un método, llamado compute, que calcula el grado de pertenencia para cada valor del universo de discurso del número difuso.

La clase FuzzySingleton implementa la interfaz MembershipFunction, que representa un valor crisp como un número difuso.

Finalmente, el fichero `systemname.java` contiene la clase que describe el sistema difuso. Esta clase es una implementación de la interfaz `FuzzyInferenceEngine`. Por tanto, los métodos públicos que implementan la inferencia son los de la interfaz (`crispInference` y `fuzzyInference`).

En este punto vamos a explicar el fichero `proyectoSI.xfl` que nos resulta de nuestro sistema `xfuzzy`.

Inicialmente nos muestra todos los `Operator set` que hemos establecido en nuestro sistema.

Como hemos visto usando el `xfuzzy` hemos establecido tres operadores con los datos siguientes:

Operador para el producto

```
operatorset producto {  
    and xfl.prod();  
    or xfl.sum();  
    not xfl.not();  
    defuz xfl.CenterOfArea();  
}
```

Operador para el mínimo

```
operatorset minimo {  
    and xfl.min();  
    or xfl.max();  
    not xfl.not();  
    defuz xfl.CenterOfArea();  
}
```

## Operador para Lucasewicz

```
operatorset lukas {  
    and xfl.lukasiewicz();  
    or xfl.max();  
    not xfl.not();  
}
```

Lo siguiente que vemos son los tipos:

El primer tipo que creamos es el tipo booleano tBool con parámetros mínimo 0.0, con máximo 1.0 y con cardinalidad 256. esto se ve entre corchetes. Y luego creamos dos Membership Function las dos trapezoidales con los valores que pone entre paréntesis, son los cuatro parámetros del trapecioide a, b, c y d.

```
type tBool [0.0,1.0;256] {  
    no xfl.trapezoid(-0.2,0.0,0.5,0.50000000000000000001);  
    si  
    xfl.trapezoid(0.5,0.50000000000000000001,1.0,1.20000000000000000002);  
}
```

Para el tipo planta creamos un tipo tPlanta con mínimo 1.0, máximo 116.0 y con cardinalidad 256. Para cada planta creamos un Membership Function de tipo singleton con valor para el parámetro a consecutivo para cada planta añadida.

Mostramos una cuantas ya que todas, sería muy pesado.

```
type tPlanta [1.0,116.0;256] {  
    Acanthaceae xfl.singleton(1.0);  
    Adiantaceae xfl.singleton(2.0);  
    Aizoaceae xfl.singleton(3.0);  
    .....  
    Violaceae xfl.singleton(114.0);
```

```

Viscaceae xfl.singleton(115.0);
Woodsiaceae xfl.singleton(116.0);
}

```

Aquí introducimos el tipo forma de pétalos, tPetalos con los valores mínimo 1.0, máximo 3.0 y de cardinalidad 256.

Añadimos tres Membership Function trapezoidales llamados libres, levemente soldados y soldados y para cada uno de ellos los valores de sus parámetros entre paréntesis.

```

type tPetalos [1.0,3.0;256] {
  libres xfl.trapezoid(0.75,1.0,1.5,1.75);
  levementeSoldados xfl.trapezoid(1.5,1.75,2.25,2.5);
  soldados xfl.trapezoid(2.25,2.5,3.0,3.25); }

```

El tipo tEstambres para el numero de estambres, poniendo sus valores mínimo a 1.0, máximo a 50.0 y cardinalidad 3.

Añadimos tres Membership Function una para pocos, otra para más o menos numerosos y finalmente otra para numerosos, con sus respectivos valores de los parámetros entre paréntesis.

```

type tEstambres [1.0,50.0;3] {
  pocos xfl.trapezoid(0.0,1.0,10.0,15.0);
  masMenosNumerosos xfl.trapezoid(5.0,10.0,30.0,35.0);
  numerosos xfl.trapezoid(25.0,30.0,49.0,50.0);
}

```

Luego se pueden ver todas las reglas para cada una de las tres reglas de inferencia pero nosotros solo mostraremos una la de rFamiliasLukas.

Aquí podemos observar las entradas que son todas las variables booleanas, 36, y dos más, la forma de pétalos y numero de estambres.

Realiza una regla para cada una de las plantas usando el operador Lukas.

En el antecedente de la regla hay una serie de and's, &, con las entradas que poseen la planta que esta en el consecuente comparándolas con el valor de dicha entrada para esa planta. En el consecuente esta la planta que cumple esas características con esos valores.

No tiene porque estar todas las entradas al sistema en la regla ya que existen algunas plantas que no poseen esa característica ni ninguno de los valores que para ella tenemos.

```
rulebase rFamiliasLukas (tBool inCaduca, tBool inPerenne, tBool inOpuesta, tBool
inAlternativa, tBool inEstipulas, tBool inEntera, tBool inDividida, tBool inSimple, tBool
inHermafrodita, tBool inUnisexual, tBool inHipogina, tBool inPerigina, tBool inEpigina,
tBool inActinomorfa, tBool inSolitaria, tBool inIrregular, tBool inEsferoidal, tBool
inGranos, tBool inSupero, tBool inInfero, tBool inBilabiada, tBool inEncapsulado, tBool
inGlanduliferos, tBool inEstrellados, tBool inArbol, tBool inArbusto, tBool inLenosa,
tBool inLiana, tBool inHerbacea, tBool inTrepadora, tBool inAcuatica, tBool inRizoma,
tBool inIsosporeo, tBool inHeterosporeo, tPetalos inNumpetalos, tEstambres inEstam :
tPlanta outPlantaLukas) using lukas {
```

```
    if(inPerenne == si & inOpuesta == si & inEstipulas == no & inSimple == si &
inHermafrodita == si & inSolitaria == si & inIrregular == si & inSupero == si &
inEncapsulado == si & inGlanduliferos == si & inArbusto == si & inEstam == pocos) ->
outPlantaLukas = Acanthaceae;
```

```
.....
.....
```

```
    if(inCaduca == si & inPerenne == si & inOpuesta == si & inEstipulas == no &
inUnisexual == si & inInfero == si) -> outPlantaLukas = Viscaceae;
    if(inPerenne == si) -> outPlantaLukas = Woodsiaceae;
}
```

Y ahora el sistema general:

```
system (tBool inCaduca, tBool inPerenne, tBool inOpuesta, tBool inAlternativa, tBool
inEstipulas, tBool inEntera, tBool inDividida, tBool inSimple, tBool inHermafrodita, tBool
```

inUnisexual, tBool inHipogina, tBool inPerigina, tBool inEpigina, tBool inActinomorfa, tBool inSolitaria, tBool inIrregular, tBool inEsferoidal, tBool inGranos, tBool inSupero, tBool inInfero, tBool inBilabiada, tBool inEncapsulado, tBool inGlanduliferos, tBool inEstrellados, tBool inArbol, tBool inArbusto, tBool inLenosa, tBool inLiana, tBool inHerbacea, tBool inTrepadora, tBool inAcuatica, tBool inRizoma, tBool inIsosporeo, tBool inHeterosporeo, tPetalos inNumpetalos, tEstambres inEstam : tPlanta outPlantaProd, tPlanta outPlantaMin, tPlanta outPlantaLukas) {

    rFamiliasProd(inCaduca, inPerenne, inOpuesta, inAlternata, inEstipulas, inEntera, inDividida, inSimple, inHermafrodita, inUnisexual, inHipogina, inPerigina, inEpigina, inActinomorfa, inSolitaria, inIrregular, inEsferoidal, inGranos, inSupero, inInfero, inBilabiada, inEncapsulado, inGlanduliferos, inEstrellados, inArbol, inArbusto, inLenosa, inLiana, inHerbacea, inTrepadora, inAcuatica, inRizoma, inIsosporeo, inHeterosporeo, inNumpetalos, inEstam : outPlantaProd);

    rFamiliasMin(inCaduca, inPerenne, inOpuesta, inAlternata, inEstipulas, inEntera, inDividida, inSimple, inHermafrodita, inUnisexual, inHipogina, inPerigina, inEpigina, inActinomorfa, inSolitaria, inIrregular, inEsferoidal, inGranos, inSupero, inInfero, inBilabiada, inEncapsulado, inGlanduliferos, inEstrellados, inArbol, inArbusto, inLenosa, inLiana, inHerbacea, inTrepadora, inAcuatica, inRizoma, inIsosporeo, inHeterosporeo, inNumpetalos, inEstam : outPlantaMin);

    rFamiliasLukas(inCaduca, inPerenne, inOpuesta, inAlternata, inEstipulas, inEntera, inDividida, inSimple, inHermafrodita, inUnisexual, inHipogina, inPerigina, inEpigina, inActinomorfa, inSolitaria, inIrregular, inEsferoidal, inGranos, inSupero, inInfero, inBilabiada, inEncapsulado, inGlanduliferos, inEstrellados, inArbol, inArbusto, inLenosa, inLiana, inHerbacea, inTrepadora, inAcuatica, inRizoma, inIsosporeo, inHeterosporeo, inNumpetalos, inEstam : outPlantaLukas);

}



Tiene como entradas todas las anteriormente descritas, 38 entradas. Y posee tres métodos borrosos que dan como resultado tres variables de salida, uno para cada uno de ellos.

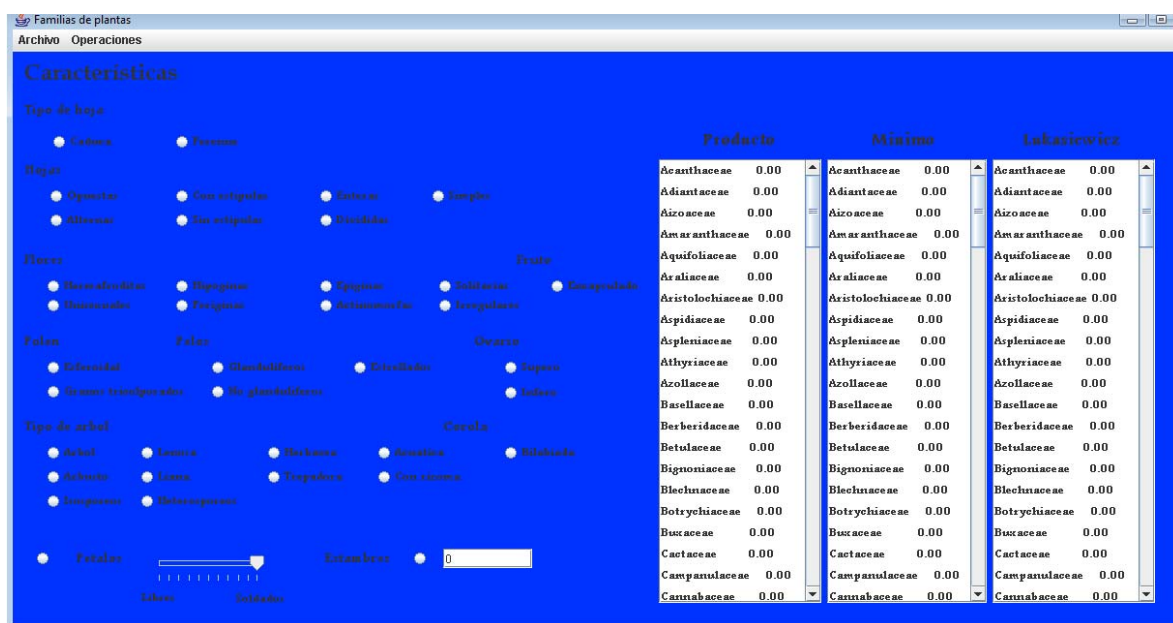
Para el método rFamiliasProd que usa el operador min la salida es outPlantaProd, para el método rFamiliasMin que usa el operador min la salida es outPlantaMin, y para el método rFamiliasLukas que usa el operador Lukas la salida es outPlantaLukas.

Ahora explicaremos lo que hemos realizado nosotras para crear una interfaz para que el usuario pueda introducir sus datos y obtener grados de pertenencia a las familias para conocer cuales son las familias que mayor probabilidad tiene de pertenecer la planta descrita a esa familia.

En esta herramienta hemos utilizado la anterior interfaz pero con modificaciones.

# EXPLICACIÓN DE LA HERRAMIENTA

Veremos a continuación una imagen con la pantalla principal de nuestra herramienta:



Aquí podemos observar como ahora aparecen todas las familias en cada una de las tres columnas de la derecha de la pantalla.

Con esta nueva herramienta nos dimos cuenta de que seria más sencillo para el usuario no tener que recordar pulsar en la barra de menú la opción de Opciones y seleccionar comenzar para poder empezar a usar la herramienta. Como esto resultaba muy difícil de recordar, en esta herramienta consideramos que al inicio de la ejecución ya apareciesen todas las familias en su respectivo lugar sin tener que realizar ninguna tarea previa.

Por lo que en un primer instante las familias ya aparecen en su lugar.

Otra novedad como se puede observa es que aparecen tres columnas todas ellas con todas las familias.

Esto es debido a que al tener que proporcionar grados de pertenencia a todas las familias y según tres reglas de inferencia, tenemos que tener tres columnas con todas las familias y todas ellas con un grado de pertenencia según los datos proporcionados por el usuario.

Con ello todas las familias inicialmente tienen grado de pertenencia 0 ya que el usuario aun no ha introducido ningún dato.

Otra novedad es la incorporación de dos nuevas formas de introducir datos.

Uno de ellos se muestra en forma de Slider y otro en forma de JtextField.

Para poder introducir estos datos para el Slider, debemos de pulsar en el RadioButton que tiene a su izquierda y entonces nos permite mover el Slider hacia la posición deseada.

Como se ve en la pantalla podemos moverlo desde libres, pasando por levemente soldados hasta soldados.

El otro debemos de introducir el número de estambres deseados y a continuación pulsar en el RadioButton que tiene a su izquierda para que la herramienta coja correctamente este valor.

Cada vez que cambiamos algo en la pantalla principal, es decir, señalamos alguna de las características o cambiamos como he dicho anteriormente alguno de los otros dos nuevos datos introducidos el grado de pertenencia cambia para las familias.

Además también proporcionamos en orden descendente, de mayor a menor, en grado de pertenencia a la familia para proporcionar al usuario una forma visual y sencilla de conocer cuales son las familias con mayor probabilidad de ser a la que pertenezca la planta a la que el usuario le esta realizando el estudio de determinación.

En esta imagen se puede observar lo descrito:

Ahora explicaremos el código de la herramienta.

Vamos a reutilizar el código de la herramienta usada para el programa nítido ya que lo que queremos conseguir es que todas las características de las plantas nos proporcionen ayuda a la hora de conseguir un grado de pertenencia a cada una de las familias.

Por ello hemos incluido en nuestro sistema Xfuzzy las 38 características, 36 características que ya teníamos en la herramienta del sistema nítido y 2 características mas que son la forma de los pétalos y el numero de estambres.

Para realizar esta herramienta borrosa hemos implementado nuestro sistema con un programa explicado anteriormente que es Xfuzzy, con este programa hemos obtenido tres clases que son interfaces y una cuarta clase que es la que implementa todas estas interfaces además de proporcionarnos un grado de pertenencia a cada una de las tres reglas de inferencia borrosa.

Ahora explicaremos el código de la clase proyectoSI.java que como hemos explicado anteriormente es la que lleva el peso de la implementación.

Inicialmente observamos como se describen todas las proposiciones básicas que relacionan una variable de entrada con una de sus etiquetas lingüísticas que proporciona por defecto el lenguaje xfl, como puede ser:

IsSmallerOrEqual, isEqual, etc.

Después vemos como aparecen cada una de las dos funciones de pertenencia que hemos utilizado en nuestro sistema que hemos realizado en Xfuzzy. Estas dos funciones son: trapezoid y singleton.

A continuación vemos como se muestra el código para los tres operator set que hemos implementado en nuestro sistema: un operador para el mínimo, otro para el producto y otro para el método lukasewicz.

Después observamos los cuatro tipos que hemos realizado en el programa Xfuzzy que implementa nuestro sistema: tBool, tPlanta, tEstambres y tPetalos.

También podemos ver como para cada una de ellas aparecen dos variables una para establecer el valor del mínimo y otra para establecer el valor del máximo.

Además se pueden observar como entre paréntesis después de cada nombre de Membership Function aparecen los valores que hemos dado a estas en el programa Xfuzzy.

También podemos ver como para el tipo tBool usa la función trapezoid al igual que el tipo tEstambres y tPetalos. Pero para el tipo tPlanta usa la función singleton.

Por ultimo mostramos las tres rulebase una para cada regla de inferencia borrosa.

RFamiliaProd para el operador producto y da como variable de salida outPlantaProd.

RfamiliaMin para el operador mínimo y da como variable de salida outPlantaMin.

RFamiliaLukas para el operador lukas (lukasewicz) y da como variable de salida outPlantaLukas.

Para cada una de ellas obtenemos por medio de un array de double que hemos creado el grado de pertenencia a cada una de las familias.

Por ejemplo para el producto y para la familia *Acanthaceae*, tenemos una regla que tiene como salida la variable double r1, en la que se guarda el valor referido a la inferencia borrosa de la regla obteniendo un grado de pertenencia de la variable \_t\_outPlantaProd.Acanthaceae (ósea de la familia *Acanthaceae*) según los valores de las variables de entrada al sistema.

Este valor lo guardamos en nuestro array de double llamado sal que es lo que devolvemos en cada función para cada rulebase.

Entonces obtenemos tres arrays de double uno para cada rulebase, ósea uno para cada regla de inferencia.

Nosotras implementamos la siguiente función: ejecutaCosa que lo que hace es guardar en un arrayList aux los tres arrays de double pertenecientes a cada una de las ejecuciones de las tres reglas de inferencia borrosa.

En la posición cero de este arrayList esta la salida de la ejecución rFamiliaProd en la que se obtiene en cada posición el grado de pertenencia de cada una de las familias según esta regla de inferencia borrosa producto.

En la posición uno de este arrayList esta la salida de la ejecución rFamiliaMin en la que se obtiene en cada posición el grado de pertenencia de cada una de las familias según esta regla de inferencia borrosa mínimo.

En la posición dos de este arrayList esta la salida de la ejecución rFamiliaLukas en la que se obtiene en cada posición el grado de pertenencia de cada una de las familias según esta regla de inferencia borrosa lukas (lukasewicz).

Es código que sigue nosotros no lo usamos pero también lo tenemos que implementar para que no de fallos al ejecutar aunque lo que hace es devolver null en caso de usarlo.

Ahora vamos a explicar como se relaciona el programa principal con esta clase que se ha creado con el Xfuzzy.

En la clase PanelPrin.java se añaden dos nuevos botones.

Añadimos tres arrays de TipoOrdenar con 116 posiciones, igual numero que el de familias, para ordenar los datos de las tres reglas de inferencia de mayor a menor, para mostrar primeramente las familias con mayor grado de pertenencia y así ir disminuyendo hasta las familias con grado de pertenencia cero.

También tenemos tres arrays de String uno para cada una de las tres reglas de inferencia con todas las familias ósea cada array de 116 String.

Además añadimos tres arrayList uno para las familias, otro para los nombres de estas y otro para la salida del xfuzzy.

Por ultimo añadimos tres variables booleanas que luego veremos lo que hacen.

En la constructora del PanelPrin se ve como se inicializan todo lo anteriormente escrito a los valores descrito, además de la booleana usando a false. También vemos que el fichero usado para obtener las características de las plantas es el plantas.txt que sigue siendo el mismo que el usado con la anterior herramienta, la nítida.

Vemos que inicializa los componentes de la interfaz grafica a continuación pero esta vez a diferencia de la anterior añadimos dos RadioButton y un Jslider y un JTextField. El Jslider es para introducir la forma de los pétalos y el JTextField para el numero de pétalos.

Además en vez de un Jlist ponemos tres uno para cada una de reglas de inferencia.

También un JLabel uno para mostrar la etiqueta Estambres y otro para la etiqueta pétalos.

Para el Jslider ponemos libres en la parte izquierda de este y a la derecha soldados. Por el centro se sobreentiende que son levemente soldados.

Después observamos en el código como para cada lista de familias, para cada una de las tres reglas de inferencia obtenemos el nombre de esta y para cada una de las familias establecemos que si la pulsamos aparezca la ventana implementada en la clase Aceptadas.java que es como la de la anterior herramienta nítida que muestra las características de la planta seleccionada.

Cuando pulsamos el RadioButton correspondiente a la etiqueta de pétalos, podemos mover el Jslider al valor deseado poniendo la variable usando a true, de tal manera que el evento del Jslider coge ese valor para cambiar los valores en las tres listas ejecutando la función ejecutar.

Además usa la función Poner2 que explicaremos a continuación.

En el JTextField introducimos el valor del numero de estambres deseado, de tal manera que cuando pulsamos en el RadioButton correspondiente a la etiqueta estambres se ejecuta la función ejecutar cambiando los valores de las tres listas además de usar la función Poner2 también.

Ahora vemos la función Poner que lo que hace es poner todos los nombres de todas las familias en cada una de las tres listas y poner un grado de pertenencia igual a 0.0.

La función Poner2 se llama cada vez que se pulsa alguna de las características que te permite seleccionar la interfaz realizada. En ella cogemos el grado de cada una de las familias en cada regla de inferencia y ponemos el nombre de las 116 familias en cada una de las tres listas con el grado que le corresponde después de ejecutar la inferencia borrosa obtenida mediante la variable SalidaXFuzzy.

Después se ordenan las familias de las tres listas para dar un resultado al usuario de mayor a menor las familias según su grado de pertenencia.

La función ordenar que ordena cada una de las tres listas según sus grados de pertenencia de mayor a menor.

La función ordenacionRapida que lo que hace es ordenar con el método quicksort.

La función Reiniciar que es la misma que en la herramienta nítida pero ahora reiniciando también para pétalos, estambres y las tres listas, en vez de una.

El método ver seleccionados igual que en la herramienta nítida.

El método ejecutar que lo que obtiene es si pulsamos alguna de las características que teníamos en la anterior herramienta nos da un grado 1.0 pero si pulsamos en las dos nuevas cosas introducidas obtenemos su valor.

Esto lo metemos a la clase que anteriormente hemos explicado y ejecutamos el método ejecutaCosa obteniendo los tres arrays con todos los grados de pertenencia a cada una de las familias.

La clase TipoOrdenar.java consta de dos variables una de tipo String para el nombre de la familia y otra de tipo double para el grado de pertenencia de las familia.

El resto de las clases no cambian sustancialmente como para ser mencionadas de nuevo.

Como los resultados no son los esperados hemos tenido que realizar un cambio en las reglas de la clase proyectoSI.java. En esta clase hemos modificado todos los métodos que nos ejecutaban las reglas que implementamos en el programa Xfuzzy.

Los resultados no son los esperados ya que si una planta después de pulsar en alguna de las características de la herramienta nítida en esta herramienta debe de proporcionar los mismos resultados si solo se pulsan las características que estaban en la herramienta nítida y no pulsar en las dos nuevas características añadidas en la herramienta borrosa.

Por ello vamos a explicar las modificaciones necesarias que hemos tenido que realizar en el código.

En los métodos RL\_rFamiliasProd, RL\_rFamiliasMin, y RL\_rFamiliasLukas son los métodos que debemos de cambiar.

Como los métodos nuevos tendrán muchas líneas hemos decidido realizar para cada uno de estos métodos 5 métodos cada uno de ellos realizara unas cuantas reglas.

El primer método en el que se divide realiza las reglas para las 27 primeras familias aunque como sabemos los arrays empiezan en la posición 0 y por lo tanto el array se rellena hasta la posición 26. Para el segundo método se ejecutan las reglas para las familias siguientes y se añaden desde la 27 hasta la 50. El tercer método ejecuta las reglas para las familias siguientes y se añaden desde la 51 hasta la 75. El cuarto método ejecuta las reglas para las familias siguientes y se añaden desde la 76 hasta la 100. Y el quinto método y ultimo para las familias siguientes y se añaden desde la 101 hasta la 115.

Obteniéndose en total 116 grados de pertenencia para las 116 familias.

En los métodos globales además de añadir las 38 variables de entrada añadimos un array de doubles noinfo que después explicaremos para que su usa.

En los métodos globales de cada uno de ellos, es decir, en los que se llaman a los cinco métodos en los que lo hemos dividido lo único que se realiza es obtener la salida en una variable de salida del método llamada sal con 116 posiciones obteniéndose un grado de pertenencia según cada operador.



Cada uno de los cinco métodos que se ejecutan en cada una de las tres bases de reglas tienen el mismo nombre que estos pero seguidos de la posición en la que se ejecutan.

El método global RL\_rFamiliasProd el primer método se llama RL\_rFamiliasProd1 y así sucesivamente hasta el quinto y ultimo llamado RL\_rFamiliasProd5.

En cada uno de estos cinco métodos introducimos el array sal y un número entero llamado j que es el que contiene la posición desde la cual debemos de añadir información al array sal.

El código que sigue es el mismo que teníamos antes de modificar la clase hasta llegar a la inicialización de todas las posiciones de tipo double del array noinfo que se le pasa como parámetro al método.

Cada una de las variables que se añaden al array serán las 38 variables de entrada al sistema cuando no se tiene información de estas.

Para distinguirlas de las variables de entrada las llamamos igual que estas pero con noInfo delante en vez de in.

Por ejemplo:

InCaduca cuando no tiene información se llama noinfoCaduca.

Estas variables tendrán un valor de uno cuando la variable de entrada inCaduca no tenga ningún valor en el sistema.

Después podemos ver cada una de las reglas de cada una de las familias.

En la anterior implementación de las bases de reglas por ejemplo para la primera familia teníamos la siguiente regla.

```
_rl =
_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op
.and(_t_inPerenne.si.isEqual(inPerenne),_t_inOpuesta.si.isEqual(inOpuesta)),_t_inEstipula
s.no.isEqual(inEstipulas)),_t_inSimple.si.isEqual(inSimple)),_t_inHermafrodita.si.isEqual(i
nHermafrodita)),_t_inSolitaria.si.isEqual(inSolitaria)),_t_inIrregular.si.isEqual(inIrregular))
,_t_inSupero.si.isEqual(inSupero)),_t_inEncapsulado.si.isEqual(inEncapsulado)),_t_inGlan
duliferos.si.isEqual(inGlanduliferos)),_t_inArbusto.si.isEqual(inArbusto)),_t_inEstam.poco
s.isEqual(inEstam));
```

```

        outPlantaProd.set(_i_outPlantaProd,_rl,                _t_outPlantaProd.Acanthaceae);
        _i_outPlantaProd++; sal [j] = _rl; j++;

```

En esta regla se pueden ver todas las características que la familia *Acanthaceae* cumple. Es decir, como la familia es perenne la variable de entrada inPerenne del sistema Xfuzzy que hemos implementado esta al valor si ya que la característica perenne la posee esta familia.

Y así con cada una de las características que cumple dicha familia.

En el caso de no tener información sobre la característica la regla producirá un valor erróneo y lo que es más importante distinto al que se genera en la herramienta nítida.

Esta regla es la que se realiza en nuestra nueva base de reglas para la primera familia de nuestro grupo de 116 familias con el operador producto.

```

_rl                                                                 =
_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op
.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.an
d(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_
op.and(_op.and(_op.and(_op.and(
        _op.or(_t_inPerenne.si.isEqual(inPerenne),noinfoPerenne),
        _op.or(_t_inCaduca.no.isEqual(inCaduca),noinfoCaduca)),
        _op.or(_t_inOpuesta.si.isEqual(inOpuesta),noinfoOpuesta)),
        _op.or(_t_inAlterna.no.isEqual(inAlterna),noinfoAlterna)),
        _op.or(_t_inEstipulas.no.isEqual(inEstipulas),noinfoEstipulas)),
        _op.or(_t_inEntera.no.isEqual(inEntera),noinfoEntera)),
        _op.or(_t_inDividida.no.isEqual(inDividida),noinfoDividida)),
        _op.or(_t_inSimple.si.isEqual(inSimple),noinfoSimple)),

_op.or(_t_inHermafrodita.si.isEqual(inHermafrodita),noinfoHermafrodita)),
        _op.or(_t_inUnisexual.no.isEqual(inUnisexual),noinfoUnisexual)),
        _op.or(_t_inHipogina.no.isEqual(inHipogina),noinfoHipogina)),
        _op.or(_t_inPerigina.no.isEqual(inPerigina),noinfoPerigina)),

```

```

        _op.or(_t_inEpigina.no.isEqual(inEpigina),noinfoEpigina)),
        _op.or(_t_inActinomorfa.no.isEqual(inActinomorfa),noinfoActinomorfa)),
        _op.or(_t_inSolitaria.si.isEqual(inSolitaria),noinfoSolitaria)),
        _op.or(_t_inIrregular.si.isEqual(inIrregular),noinfoIrregular)),
        _op.or(_t_inEsferoidal.no.isEqual(inEsferoidal),noinfoEsferoidal)),
        _op.or(_t_inGranos.no.isEqual(inGranos),noinfoGranos)),
        _op.or(_t_inSupero.si.isEqual(inSupero),noinfoSupero)),
        _op.or(_t_inInfero.no.isEqual(inInfero),noinfoInfero)),
        _op.or(_t_inBilabiada.no.isEqual(inBilabiada),noinfoBilabiada)),
        _op.or(_t_inEncapsulado.si.isEqual(inEncapsulado),noinfoEncapsulado)),

_op.or(_op.or(_t_inGlanduliferos.si.isEqual(inGlanduliferos),_t_inGlanduliferos.no.isEqual
(inGlanduliferos)),noinfoGlanduliferos)),
        _op.or(_t_inEstrellados.no.isEqual(inEstrellados),noinfoEstrellados)),
        _op.or(_t_inArbol.no.isEqual(inArbol),noinfoArbol)),
        _op.or(_t_inArbusto.si.isEqual(inArbusto),noinfoArbusto)),
        _op.or(_t_inLenosa.no.isEqual(inLenosa),noinfoLenosa)),
        _op.or(_t_inLiana.no.isEqual(inLiana),noinfoLiana)),
        _op.or(_t_inHerbacea.no.isEqual(inHerbacea),noinfoHerbacea)),
        _op.or(_t_inTrepadora.no.isEqual(inTrepadora),noinfoTrepadora)),
        _op.or(_t_inAcuatica.no.isEqual(inAcuatica),noinfoAcuatica)),
        _op.or(_t_inRizoma.no.isEqual(inRizoma),noinfoRizoma)),
        _op.or(_t_inIsosporeo.no.isEqual(inIsosporeo),noinfoIsosporeo)),

_op.or(_t_inHeterosporeo.no.isEqual(inHeterosporeo),noinfoHeterosporeo)),
        _op.or(0,noinfoNumpetalos)),
        _op.or(_t_inEstam.pocos.isEqual(inEstam),noinfoEstam));

outPlantaProd.set(_i_outPlantaProd,_rl, _t_outPlantaProd.Acanthaceae);
_i_outPlantaProd++;
sal [j] = _rl; j++;

```

En esta regla se puede ver que las características que teníamos en la anterior regla comprobando su igualdad con el valor del tipo tBool a si, la tenemos en esta misma regla pero realizando un or con la variable noinfo de la característica.

Para el resto de las características que no estaban en la regla realizamos el operador or cuyo primer miembro se comprueba su igualdad con el valor del tipo tBool a no y cuyo segundo miembro es la variable noinfo de la característica.

Para los tipos que no son tBool que son tEstambres y tPetalos la operación or se realiza de distinto modo.

Para la variable de entrada al sistema inNumpetalos se realiza la or entre el valor cero y noinfo de la misma característica (noinfoNumpetalos) cuando en la familia esta característica no se posee.

Cuando la familia si posee esta característica se debe de realizar la operación or entre la comprobación con el valor que posee la familia de la característica (por ejemplo libres) y la variable noinfo de la misma característica.

Para la variable de entrada al sistema inEstambres se realiza la or entre la comprobación si el valor de esta variable es igual al valor que posee la familia de esta característica (en este caso pocos) y la variable noinfo de la misma característica. Esto es para cuando esta característica si que la posee la familia.

Cuando la familia no posee esta característica la operación or se realiza entre el valor cero y noinfo de la misma característica.

Esto se realiza para todas las familias y en cada uno de los tres métodos globales de las bases de reglas.

De cada método de cada uno de los cinco en los que se divide cada una de las bases de reglas devolvemos el array sal que como hemos visto después cada regla se va metiendo un valor y así hasta que se rellenen las 116 posiciones del array.

En el primer método se rellenan las primeras 26 posiciones y así se siguen rellinando las siguientes posiciones en los siguientes métodos, hasta que en el método final, el quinto, se rellenan las 116 posiciones.

Por ello introducimos la variable global j en cada uno de los cinco métodos para continuar introduciendo los valores en las posiciones sucesivas.

Esto se realiza así ya que en un mismo método no nos dejaba introducir el gran número de líneas que se tendrían que introducir si se juntasen los cinco métodos. El programa no nos lo permitirían.

El método global RL\_rFamiliasProd nos devuelve el array de doubles con todos los valores ya introducidos en el array que se guardan los valores que resultan de la ejecución de cada una de las reglas.

Para los tres métodos globales para cada una de las bases de reglas para cada uno de los tres operadores se realiza lo mismo anteriormente explicado.

El siguiente método crispInference2 es parecido al método ejecutaCosa que realizamos en la anterior implementación.

Solo se diferencia en que como parámetros se introduce un array de doubles noinfo que lo que nos va a dar será como resultado de la salida de la ejecución de las reglas noinfo en el caso de que no haya información de ninguna de las características de las familias.

Después introducimos en el array aux en la primera posición un array de 116 posiciones con la ejecución de la base de reglas para el producto. La segunda posición del array aux un array de 116 posiciones con la ejecución de la base de reglas para el mínimo. En la tercera posición del array aux un array de 116 posiciones con la ejecución de la base de reglas para el método lukasewicz.

En la clase PanelPrin.java cambiamos varias cosas.

En lo implementado anteriormente existían tres cuadros de texto uno para cada una de las tres bases de reglas. Ahora solo existe un cuadro de texto en los que se van a introducir en una misma línea los valores de cada una de las tres bases de reglas.

## NOVEDAD EN LA HERRAMIENTA

La etiquetas con Producto, mínimo y Lukasewicz también se eliminan y se pone la única etiqueta Mínimo < Producto < Lukasiewicz.

Los eventos de los dos elementos gráficos eliminados los dos Jlist, los dos cuadros de texto eliminados también deben de ser eliminados.

Para cada uno de los eventos de los botones y de los dos elementos gráficos que añadimos en la herramienta borrosa se cambia lo que ejecutan en su cuerpo.

Se ejecuta solo una vez el método Poner2 ya que como hemos mencionado anteriormente solo vamos a poner las familias en un cuadro de texto y no en tres como en la anterior implementación.

El método Poner va a poner en el cuadro de texto Jlist1 el nombre de la familia unos cuantos espacios en blanco y después inicializamos el valor del grado de pertenencia a noinfo, en vez de 0.00 como en la anterior implementación.

El método Poner2 lo que realiza es coger la salida de la ejecución de las tres bases de reglas en la primera posición del Array SalidaXFuzzy tenemos todos los grados de pertenencia de las 116 familias con la base de reglas con el operador producto. La segunda posición con el operador mínimo y la tercera posición con el operador del método lukasewicz.

Entonces cogiendo para cada posición de cada uno de las tres posiciones del array se construye un String con los valores para cada una de las familias de cada uno de los tres operadores.

El string tiene el nombre de la familia y después el grado de pertenencia con el operador mínimo, seguido del producto y por ultimo el de lukasewicz.

Después ordenamos estos valores por el valor del grado de pertenencia que da con el operador mínimo y colocamos lo que resulta en cuadro de texto de la derecha de la herramienta borrosa.

Este método se ejecuta cada vez que pulsamos en alguno de los elementos gráficos usados para introducir las características de las familias.

En el método ordenar ya solo se ordena una sola vez no hace falta distinguir entre los tres cuadros de texto ya que ahora solo tenemos un solo cuadro de texto por el que ordenaremos por la columna de grados de pertenencia por el operador del mínimo.

El método ordRapida y verSeleccionados no cambian.

El método ejecutar que se ejecuta cada vez que pulsamos en alguno de los elementos gráficos usados para introducir las características de las familias también es distinto porque tenemos que introducir el array noinfo.

Este se inicializa a un valor de cero.

El valor de los elementos gráficos seleccionados pone el valor de la variable xfuzzy a uno.

El valor de los elementos gráficos que no están seleccionados ponemos el valor de la variable noinfo a uno.

Ya que el grado de pertenencia en los elementos boléanos es de cero si el elemento no esta pulsado y uno si esta pulsado.

Para los otros dos elementos gráficos también conseguimos como en la anterior implementación un valor de cero si no esta seleccionado y el valor introducido si esta seleccionado.

Después ejecutamos el método crispInference2 y se guarda en el ArrayList SalidaXFuzzy. Por lo tanto este ArrayList tiene tres posiciones con tres arrays en cada una de las posiciones que es de 116 posiciones que contiene el grado de pertenencia de cada una de las tres bases de reglas una para cada uno de los operadores implementados en la herramienta Xfuzzy.

Al método crispInference2 se le introduce el array xfuzz y el array noinfo para poder ejecutar las reglas correctamente.

Como la herramienta muestra un solo cuadro de texto con los tres grados de pertenencia a los tres métodos de inferencia borrosa ordenadas por el grado del producto, decidimos descartar los métodos de inferencia y quedarnos solo con el producto.

Ya que el método del mínimo en cuando tenemos un valor cero en alguna de las premisas el valor de salida es cero, produciéndonos una salida con muy poco información relevante.

El método de lukasewicz tampoco nos aporta gran información ya que la salida proporciona información confusa.

Por estas razones vamos a explicar lo que hemos cambiado en la herramienta además de realizar una herramienta más interactiva.

Las razones de que sea más interactiva es que los botones puedan ser seleccionados y deseleccionados. También proporcionamos al igual que en la herramienta nítida una opción para ver las familias descartadas y poder conocer las razones de porque han sido descartadas.

Explicaremos ahora lo añadido o cambiado en las distintas clases, lo que no haya sido cambiado o modificado no será explicado nuevamente.

En la clase Porque.java hemos añadido que al cuadro de texto que aparece en la ventana donde se muestran las razones de porque una familia es descartada no es posible introducir texto ya que hemos inhabilitado esta opción.

En el método Escribir de esta clase hemos introducido que delante de cada razón de descarte ponga un guión y después dos líneas para separar las distintas razones entre si.

En la clase Descartados.java hemos añadido dos nuevos arrays un ArrayList para la información de la familia y otro array para el grado de pertenencia de la familia.

En la constructora de esta clase se le introduce un ArrayList que se lo asignamos al ArrayList del que hemos hablado anteriormente para la información de la familia.

En el evento que se ejecuta cuando pulsamos algún elemento de la lista del cuadro de texto, es decir, si pulsamos en alguna de las familias extraemos el numero de línea y miramos en el ArrayList de familias en la posición 0 de la fila que corresponda con la pulsada obteniendo así el nombre de la familia pulsada guardando dicha información en la variable global nombre para conservarlo para otros métodos.

Cuando pulsamos en el menú de opciones Razones Descartadas cogemos del ArrayList información de familias el nombre el nombre de la familia y seleccionamos las razones de descarte para ponerlas en el cuadro de texto de la ventana que aparece. Realizando esto mediante el método Escribir de la clase Porque.java explicado anteriormente.



En vez del método Poner ahora Poner realizamos otro método que se llama muestraDescartados al que se le pasa el array de familias y el array de grados de pertenencia.

Cuando el grado de pertenencia de una familia es igual a cero entonces se añade a la lista de familias descartadas, la cual es pasada al elemento del cuadro de texto donde se muestra la lista de familias descartadas.

En la clase PanelPrin.java quitamos dos de los tres paneles ya que solamente vamos a mostrar las tres reglas en una sola columna.

Además permitimos que el usuario pueda seleccionar y deseleccionar todos los botones de la pantalla principal para hacer que la herramienta sea más interactiva.

Todos los eventos llaman al método mostrarGrados en vez de al método Poner2 como hacíamos anteriormente.

Al método Poner lo hemos renombrado llamándolo inicializaGrados que es realmente lo que realiza este método.

Después el método mostrarGrados que lo que realiza es lo mismo que hacia el método Poner.

Después el método ordRapida ordenamos solo para el producto.

El resto de métodos no comentados siguen igual.

En la clase proyectoSI.java solo dejamos la inferencia para la base de reglas para el producto teniendo en cuenta que para realizar todas las reglas una para cada una de las 116 familias de nuestro sistema debemos de seguir teniendo 5 métodos que realicen todas algunas reglas de las familias para después añadirlas en orden al array de salida sal del método RL\_rFamiliasProd que nos devuelve el array de grados de pertenencia de esa familia una vez ejecutada la regla de inferencia borrosa correspondiente.

Ahora introducimos un nuevo método Descartados para que la herramienta borrosa pueda generar los mismos resultados que la nítida, entre ellos la posibilidad de conocer las familias que están descartadas que en este caso como no se eliminan de la pantalla principal, son las que tienen grado cero.

Al ser demasiadas líneas para un solo método hemos de ponerlo dividido en 5 métodos los cuales ejecutan en orden lo relacionado con cada familia y luego lo concatenan en el ArrayList de salida del método.

En cada uno de los métodos rellenamos el array de enteros motivos que es el que va a contener los motivos porque una familia esta descartada.

En el caso de que una determinada característica de la planta tenga el valor si entonces no hay motivo de descarte y este se le asigna el valor cero. En el caso de que tenga una determinada característica a no debemos añadirle al array motivos el numero del motivo que corresponda.

Por ejemplo:

```
if(_op.or( _t_inPerenne.si.isEqual(inPerenne),noinfoPerenne)==0)motivos.add(0);
```

La primera familia posee la característica Perenne por lo que añadimos al array motivos un cero.

```
if(_op.or( _t_inCaduca.no.isEqual(inCaduca),noinfoCaduca)==0) motivos.add(1);
```

Como no posee la característica Caduca añadimos al array de motivos el numero que corresponde con que la familia no tiene la propiedad Caduca entre sus características.

En el método `crispInference2` añadimos la salida de los motivos de descarte de las familias, para devolvérselos a la clase Princip.java.

En la clase Princip.java en su constructora llamamos al método `inicializarGrados` que lo que realiza es lo mismo que el método `Poner`, coloca en el cuadro de texto de la pantalla principal el nombre de todas las familias y un texto `noinfo` ya que no hemos empezado a usar la herramienta aun.

En todos los métodos donde ponía `Poner` debemos de escribir `inicializarGrados` como en el evento del menú de opciones `Reiniciar`.

Como modificación en el cuadro de texto solo se mostrara el grado de pertenencia de las familias con respecto a la regla de inferencia del producto.

# EJECUCIÓN DE LA HERRAMIENTA

Inicialmente se muestra esta pantalla:

**Familias de plantas**

Archivo Operaciones

### Características

**Tipo de hoja**

☐ Caduca ☐ Perenne

**Hojas**

☐ Opuestas ☐ Con estípulas ☐ Enteras ☐ Simples

☐ Alternas ☐ Sin estípulas ☐ Divididas

**Flores**

☐ Hermafroditas ☐ Hipognas ☐ Tipognas ☐ Solitarias

☐ Unisexuales ☐ Perigonas ☐ Actinomorfas ☐ Regulares

**Fruto**

☐ Encapsulado

**Polen**

☐ Estrobilado ☐ Glandular ☐ Estrobilado

☐ Grano tricolporado ☐ No glandular

**Ovario**

☐ Supero ☐ Infero

**Tipo de árbol**

☐ Arbol ☐ Leñoso ☐ Herbáceo ☐ Acuático

☐ Arbusto ☐ Liana ☐ Trepadora ☐ Con rizoma

☐ Escandido ☐ Heterospermo

**Cereza**

☐ Glabra

**Pétalos**

☐ Solitarios ☐ Llanos

**Estambres**

☐ 0

**Grado de pertenencia**

Acanthaceae	no info
Adiantaceae	no info
Aizoaceae	no info
Amaranthaceae	no info
Aquifoliaceae	no info
Araliaceae	no info
Aristolochiaceae	no info
Aspidiaceae	no info
Aspleniaceae	no info
Athyriaceae	no info
Azollaceae	no info
Basellaceae	no info
Berberidaceae	no info
Betulaceae	no info
Bignoniaceae	no info
Blechnaceae	no info
Botrychiaceae	no info
Buxaceae	no info
Cactaceae	no info
Campanulaceae	no info
Cannabaceae	no info
Capparaceae	no info
Caryophyllaceae	no info
Casuarinaceae	no info
Celastraceae	no info
Ceratophyllaceae	no info
Chenopodiaceae	no info
Cistaceae	no info
Cornaceae	no info
Crasulaceae	no info
Cryptogrammaceae	no info

Al principio salen todas las familias con un grado de pertenencia para el que no tenemos información acerca de la pertenencia ya que en este primer instante no existe ninguna información al respecto.

Cuando pulsamos alguno de los elementos gráficos que estaban en la herramienta nítida cambia la información acerca del grado de pertenencia de las familias.

El grado de pertenencia se mostrara de este modo:

The screenshot shows the 'Familias de plantas' application window. The 'Características' section on the left contains various input fields for plant characteristics, including 'Tipo de hoja', 'Hojas', 'Flores', 'Polen', 'Tipo de árbol', and 'Fruto'. The 'Grado de pertenencia' list on the right shows a scrollable list of plant families, all with a membership degree of 1.00. The 'Fruto' field is set to 'Encapsulado'.

Familia	Grado de pertenencia
Paeoniaceae	1,00
Papaveraceae	1,00
Phytolaccaceae	1,00
Osmundaceae	1,00
Ophioglossaceae	1,00
Orchidaceae	1,00
Orobanchaceae	1,00
Polygonaceae	1,00
Polypodiaceae	1,00
Portulacaceae	1,00
Plumbaginaceae	1,00
Pinaceae	1,00
Pitosporaceae	1,00
Platanaceae	1,00
Lythraceae	1,00
Malvaceae	1,00
Marsileaceae	1,00
Lycopodiaceae	1,00
Lauraceae	1,00
Leguminosae	1,00
Lentibulariaceae	1,00
Myrtaceae	1,00
Nyctaginaceae	1,00
Onagraceae	1,00
Myoporaceae	1,00
Martyniaceae	1,00
Molluginaceae	1,00
Moraceae	1,00
Theligonaceae	1,00
Thelypteridaceae	1,00
Thymelaceae	1,00

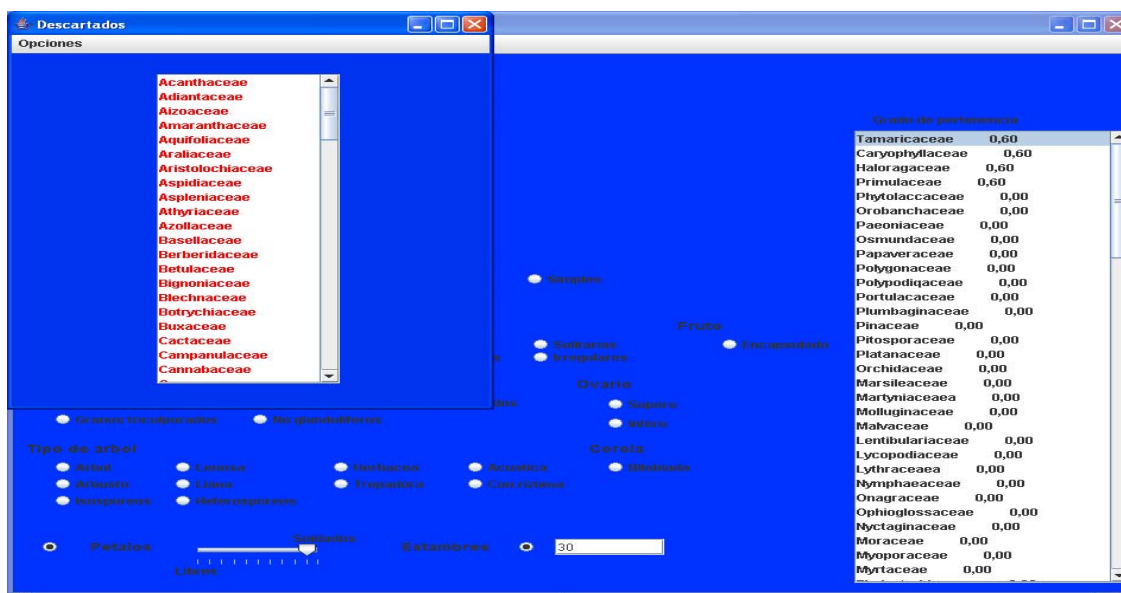
Las familias que tienen grado de pertenencia 1 es que estas familias son candidatas a ser la familia a la que pertenece la planta a determinar.

Las familias que tienen grado cero según las características que ha señalado el usuario y después de haber realizado la inferencia borrosa, entonces aparecen estas familias cuando pulsas la opción Descartados del menú de Opciones.

The screenshot shows the 'Familias de plantas' application window. The 'Características' section on the left is the same as in the previous screenshot. The 'Grado de pertenencia' list on the right now shows a scrollable list of plant families, all with a membership degree of 0.00. The 'Fruto' field is set to 'Encapsulado'.

Familia	Grado de pertenencia
Tamaricaceae	0,00
Caryophyllaceae	0,00
Haloragaceae	0,00
Primulaceae	0,00
Phytolaccaceae	0,00
Orobanchaceae	0,00
Paeoniaceae	0,00
Osmundaceae	0,00
Papaveraceae	0,00
Polygonaceae	0,00
Polypodiaceae	0,00
Portulacaceae	0,00
Plumbaginaceae	0,00
Pinaceae	0,00
Pitosporaceae	0,00
Platanaceae	0,00
Orchidaceae	0,00
Marsileaceae	0,00
Martyniaceae	0,00
Molluginaceae	0,00
Malvaceae	0,00
Lentibulariaceae	0,00
Lycopodiaceae	0,00
Lythraceae	0,00
Nymphaeaceae	0,00
Onagraceae	0,00
Ophioglossaceae	0,00
Nyctaginaceae	0,00
Moraceae	0,00
Myoporaceae	0,00
Myrtaceae	0,00

Entonces aparecerá esta pantalla con las opciones correspondientes como en la herramienta nítida.



Cuando pulsas algunas de las opciones para la inferencia no solo con características booleanas sino con características borrosas como son la descripción de los pétalos y el numero de los estambres, se ve como cambian los grados de pertenencia no solo se muestran valores de ceros y unos sino números intermedios debido a que estos elementos gráficos tienen valores intermedios para generar las reglas de inferencia.

Cuando pulsamos en el botón que aparece a la izquierda de la etiqueta que pone pétalos nos permite cambiar con el Jslider la descripción de estos para nuestra planta. Podemos cambiarlo moviendo dicho elemento grafico como muestra esta imagen.

**Familias de plantas**

Archivo Operaciones

### Características

**Tipo de hoja**

☐ Caduca ☒ Perenne

**Hoja**

☐ Opuestas ☐ Con estípulas ☐ Enteras ☐ Simple

☐ Alternas ☐ Sin estípulas ☐ Divididas

**Flora**

☐ Hermatroditas ☐ Hipognas ☐ Epiginas ☐ Solitarias ☐ Encapsuladas

☐ Unisexuales ☐ Periginas ☐ Actinomorfas ☐ Bregolares

**Polen** **Petalos** **Ovario**

☐ Estrobilado ☐ Glandíferos ☐ Estrobilado ☐ Supero

☐ Granos tricolporados ☐ No glandíferos ☐ Infero

**Tipo de árbol** **Cerola**

☐ Arbol ☐ Lenoso ☐ Herbacea ☐ Acuatica ☐ Dividido

☐ Arborescente ☐ Liana ☐ Trepadora ☐ Con raizena

☐ No perenne ☐ Heteropereos

**Petalos** ☐ Petalos ☐ Estambres ☐ 0

**Grado de pertenencia**

Grossulariaceae	0,60
Guttiferae	0,60
Haloragaceae	0,60
Elatinaceae	0,60
Caryophyllaceae	0,60
Celastraceae	0,60
Cornaceae	0,60
Hemionitidaceae	0,60
Trapaceae	0,60
Ulmaceae	0,60
Violaceae	0,60
Tamaricaceae	0,60
Lauraceae	0,60
Papaveraceae	0,60
Primulaceae	0,60
Aquifoliaceae	0,60
Araliaceae	0,60
Polygonaceae	0,00
Plumbaginaceae	0,00
Pinaceae	0,00
Pilosporaceae	0,00
Platanaceae	0,00
Psilotaceae	0,00
Pteridaceae	0,00
Punicaceae	0,00
Polypodiaceae	0,00
Portulacaceae	0,00
Proteaceae	0,00
Nyctaginaceae	0,00
Nymphaeaceae	0,00
Onagraceae	0,00

Después para poder introducir datos en el numero de los estambres, inicialmente debemos escribir el numero de estambres y después pulsar el botón que aparece a la derecha de la etiqueta Estambres. Se puede ver como cambia el grado de pertenencia de las familias mostradas.

**Familias de plantas**

Archivo Operaciones

### Características

**Tipo de hoja**

☐ Caduca ☒ Perenne

**Hoja**

☐ Opuestas ☐ Con estípulas ☐ Enteras ☐ Simple

☐ Alternas ☐ Sin estípulas ☐ Divididas

**Flora**

☐ Hermatroditas ☐ Hipognas ☐ Epiginas ☐ Solitarias ☐ Encapsuladas

☐ Unisexuales ☐ Periginas ☐ Actinomorfas ☐ Bregolares

**Polen** **Petalos** **Ovario**

☐ Estrobilado ☐ Glandíferos ☐ Estrobilado ☐ Supero

☐ Granos tricolporados ☐ No glandíferos ☐ Infero

**Tipo de árbol** **Cerola**

☐ Arbol ☐ Lenoso ☐ Herbacea ☐ Acuatica ☐ Dividido

☐ Arborescente ☐ Liana ☐ Trepadora ☐ Con raizena

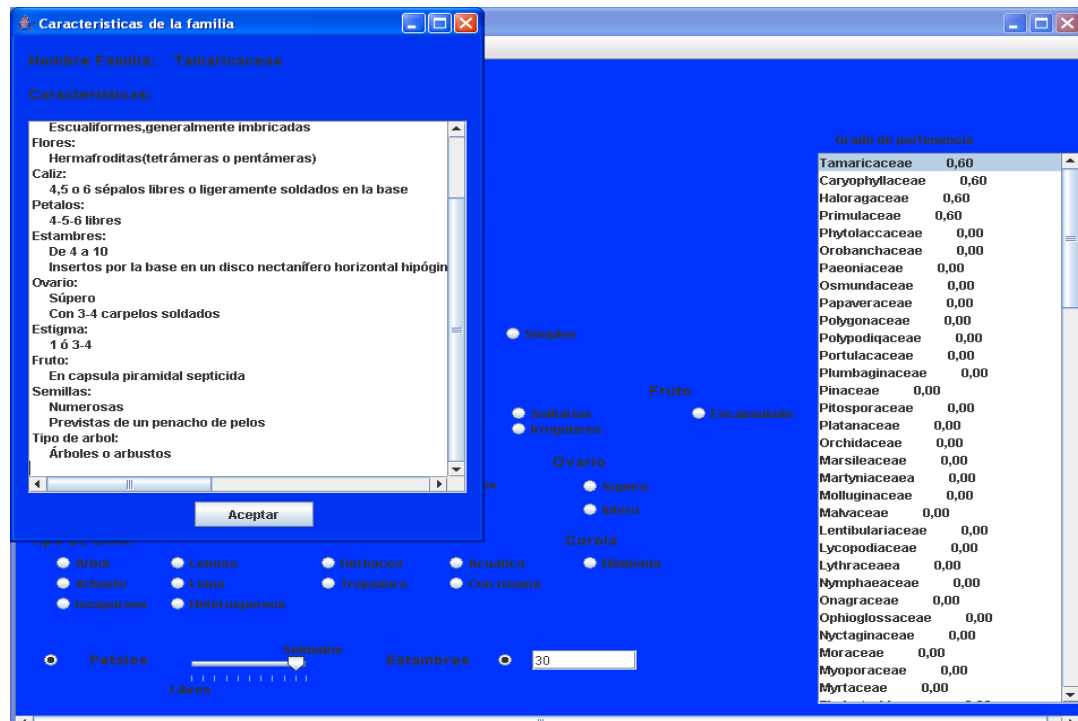
☐ No perenne ☐ Heteropereos

**Petalos** ☐ Petalos ☐ Estambres ☐ 30

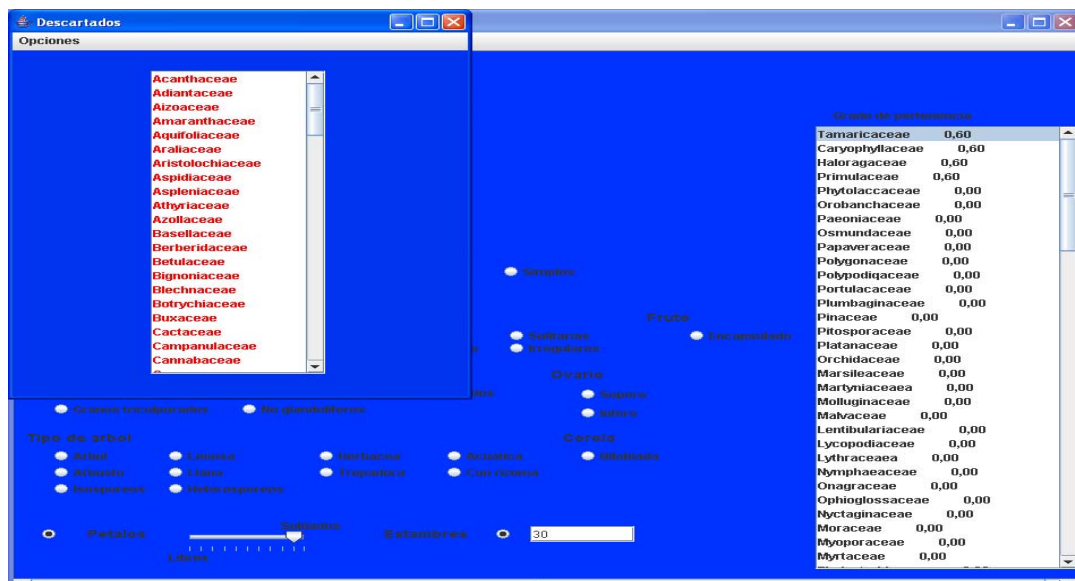
**Grado de pertenencia**

Tamaricaceae	0,60
Caryophyllaceae	0,60
Haloragaceae	0,60
Primulaceae	0,60
Phytolaccaceae	0,00
Orobanchaceae	0,00
Paeniaceae	0,00
Osmundaceae	0,00
Papaveraceae	0,00
Polygonaceae	0,00
Polypodiaceae	0,00
Portulacaceae	0,00
Plumbaginaceae	0,00
Pinaceae	0,00
Pilosporaceae	0,00
Platanaceae	0,00
Orchidaceae	0,00
Marsileaceae	0,00
Martyniaceae	0,00
Molluginaceae	0,00
Mabaceae	0,00
Lentibulariaceae	0,00
Lycopodiaceae	0,00
Lythraceae	0,00
Nymphaeaceae	0,00
Onagraceae	0,00
Ophioglossaceae	0,00
Nyctaginaceae	0,00
Moraceae	0,00
Myoporaceae	0,00
Myrtaceae	0,00

Para mostrar las características de las familias basta con pulsar en la familia deseada para ver la información.



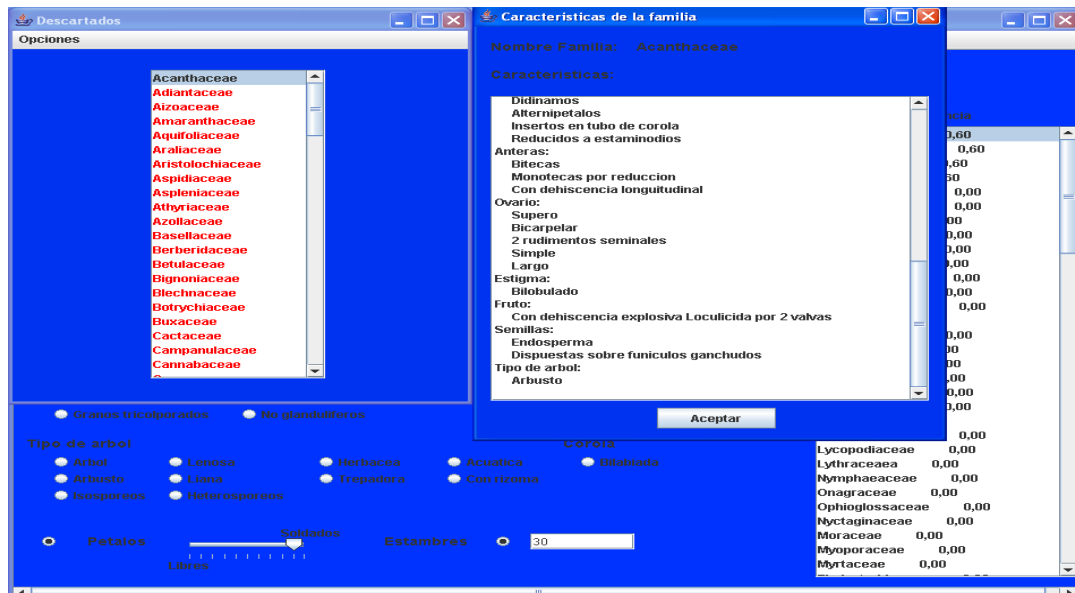
Después para ver las familias descartadas se pulsa en la opción Descartados del menú Opciones de la pantalla principal.



Como en anteriores herramientas pulsando en cualquiera de ellas podemos ver tanto las características de dicha familia como las razones del descarte, en nuestro caso en esta lista se mostraran las familias con grado de pertenencia cero obtenido mediante la aplicación de la regla de inferencia borrosa del producto.

Como en la herramienta nítida cuando pulsamos alguna de las opciones del menú Opciones aparecen las distintas ventanas con esta información.

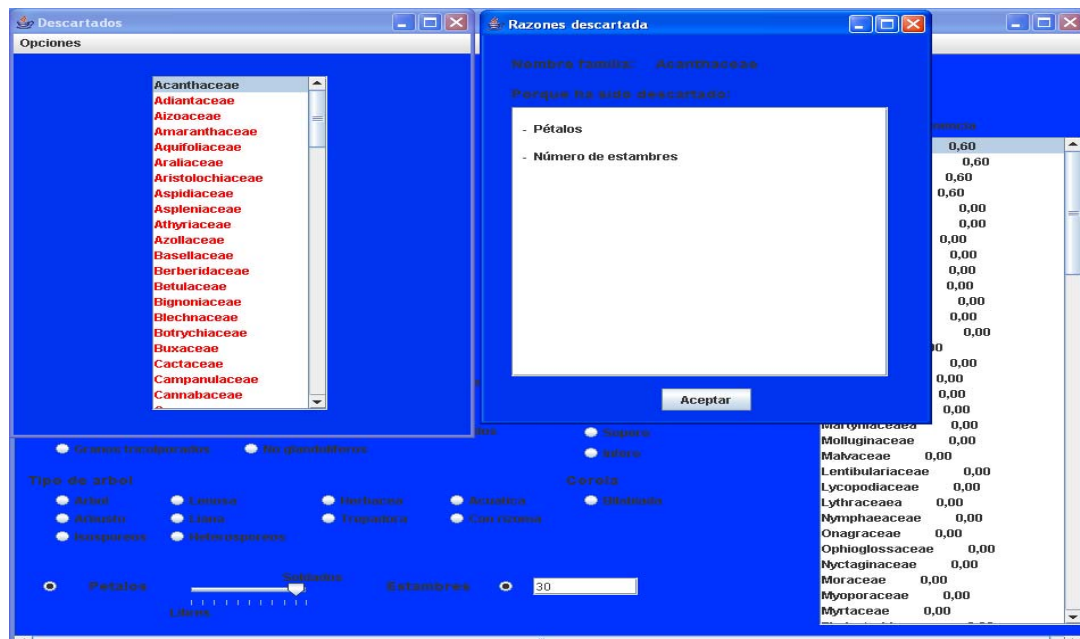
Si pulsamos en la opción Características se muestra una ventana como esta:



Para cerrar la ventana en la que aparecen las características de la familia debemos de pulsar en el botón Aceptar para una buena ejecución del programa.

Cuando pulsas en Razones descartadas aparece la siguiente ventana.

Para cerrar esta ventana con las razones de descarte pulsamos en el botón Aceptar para el buen funcionamiento de la herramienta.





Para cerrar la ventana basta con pulsar en la opción Cerrar del menú de opciones de la ventana que inicialmente ha salido cuando hemos pulsado en la opción Descartados del menú Opciones de la pantalla principal de esta herramienta borrosa.

# **HERRAMIENTA BASADA EN EL CONOCIMIENTO DEL EXPERTO**

Nuestro experto nos indico las características que le gustaría que añadiésemos en la herramienta, pero al proporcionarnos muchas características nuevas y muchas de ella dependían de si otra característica esta seleccionada decidimos hacer una nueva herramienta.

Por ello realizamos una nueva herramienta que tiene en su parte derecha un cuadro de texto que mostrara la lista de las plantas. En la parte izquierda de la pantalla se mostraran las distintas pestañas que representaran las distintas partes de las familias que el experto biólogo nos indico.

Cuando pulsas alguna de las características que conllevan la selección de otras características, estas características aparecen en la pantalla para poder ser pulsadas.

En caso de no pulsar aquellas características que conllevarían la selección de otra característica para determinar correctamente la familia, estas características no serán visibles para el usuario.

Las razones de descarte y la aparición de la descripción de las distintas familias se realiza del mismo modo que en anteriores herramientas explicadas.

La aparición de las familias cuando se inicia la ejecución de la herramienta también esta introducido en esta ultima herramienta.

En esta herramienta es indiferente el pulsar la cruz para cerrar las ventanas o pulsar en su correspondiente opción de cerrar.

En esta herramienta el experto biólogo nos proporcionó una serie de características que debíamos de introducir en ella ya que con dichas características conseguiríamos una determinación de la familia en la que se encuentra la planta que estamos introduciendo.

En este documento de Excel que se adjunta se puede ver se pueden ver dos hojas.

En la primera de las hojas del documento de Excel se aprecia como en la parte izquierda están las distintas partes de las flores que debemos colocar en cada una de las pestañas de nuestra herramienta. Estas partes están en negrita.

También se pueden ver que las características que aparecen debajo son las características que pondremos en cada pestaña, cada una con su correspondiente botón.

Cuando vemos que pone si tiene alguna de las características y aparecen características debajo de esta, dichas características serán visibles al usuario si se ha pulsado la correspondiente característica de la que dependen anteriormente.

Si dicha característica no es pulsada el usuario no vera los botones de las características que aparecen en el documento de Excel debajo de dicha condición de la característica.

En la siguiente hoja de Excel vemos las características borrosas que se podrían añadir a la herramienta y que en esta herramienta no están implementadas. Se dejara abierto entonces este proyecto para su posible ampliación con las características borrosas. Ya que con el manual proporcionado del programa Xfuzzy y explicando los distintos pasos que nosotras hemos realizado será fácil poder continuar su ampliación.

Ahora explicare mas a fondo la herramienta con las distintas clases en Java que hemos creado.

# EXPLICACIÓN DE LA HERRAMIENTA BASADA EN EL CONOCIMIENTO DEL EXPERTO

Empezaremos explicando uno de los documentos de texto que es necesario conocer.

El documento de texto Características.txt se debe de rellenar de la siguiente forma.

En el comienzo de cada una de las líneas del documento pondremos el nombre de la familia y a continuación todos y cada uno de los grados de pertenencia de esta familia a las características añadidas en esta herramienta.

Como en esta herramienta vamos a indicar la suma de todos los grados de las características que el usuario pulse en ella debemos de tener en un documento como éste, todos y cada uno de los grados de todas las características que se puedan pulsar.

En nuestro caso son 206 características por lo que cada una de las familias debe de tener una ristra de números en la línea a continuación del nombre de la familia de 206 números.

Es importante mencionar que si no están las 206 posiciones rellenas con el grado de pertenencia de dicha familia a esa característica la herramienta no funcionara.

Estos números indican:

-1 para las familias que sean incompatibles con la característica

0 para las familias que esa característica sea nula

1-3 según sea de relevante la característica en la familia

Teniendo ya todo el documento de texto terminado es posible realizar la determinación de familias de plantas.

Una peculiaridad de esta herramienta son los JComboBox creados por lo que en el documento de texto en la posición que corresponda para indicar esta característica en la familia debemos de leernos el documento de texto Ayuda.txt en el que nos dirá exactamente que poner en dicha posición.

En el documento de texto Ayuda.txt se explican las diferentes opciones que podemos introducir en cada una de las posiciones de cada una de las líneas después de escribir el nombre de la familia.

Cada una de las posiciones de la línea que se deben introducir justo a continuación del nombre de la familia están explicadas en dicho documento.

Por ejemplo para Pétalos podemos poner 3, 4, 5, Muchos o Variable y en dicho documento lo debemos de rellenar exactamente igual, con las mayúsculas correspondientes, que aparece Ayuda.txt. En el que nos dice como se debe de rellenar el documento de texto. Nos explica las diferentes opciones que se pueden elegir para cada una de las posiciones de la línea con la lista de números correspondientes al grado de pertenencia de cada una de las familias a la característica correspondiente.

Por mencionar alguna, el grado de pertenencia que se puede poner a la característica Biotipo->Árbol que es el primer numero que aparece después del nombre de la familia varia desde -1 hasta 3.

Para algún Jcombobox como es Pétalos->Numero podemos poner 3, 4, 5, Variable o Muchos.

Así explicamos las distintas opciones que podemos introducir en cada una de las características aunque es necesario mencionar y recalcar la importancia de la introducción de los 206 números pertenecientes a los 206 grados de pertenencia de todas y cada una de las 206 características de las familias.

Otro documento de texto es Ayuda2.txt en el que nos dice como rellenar el documento de texto Características.txt. Nos indica que debemos de poner el nombre de familia y a continuación los 206 grados de pertenencia, dejando un espacio en blanco después del nombre de la familia y otro después de cada grado.

Cada vez que terminamos una línea debemos de pulsar el retorno de carro (intro).

Se pueden escribir en dicho documento las familias que se desee.

Por ultimo nos indica que debemos de introducir todas y cada una de las 206 características ya que sino la herramienta no funcionara. Esto es muy importante y es un dato a tener en cuenta.

Después de explicar todos los documentos de texto empezaremos explicando las clases Java.

La clase Ventana.java es la que inicialmente aparece cuando se ejecuta la herramienta.

En ella podemos apreciar como el titulo es Determinación de familias de plantas.

También se puede ver como añadimos los distintos menús de opciones a la barra de herramientas.

En el menú Menu tenemos la opción Salir.

En el menú Operaciones la única opción que tenemos es la de Reiniciar.

Por ultimo en el menú de Ayuda tenemos las opciones Rellenar características.txt, Instrucciones de uso y Acerca del programa.

Explicaremos a continuación los distintos eventos que realizan todas las opciones de los distintos menús.

Cuando pulsamos en la opción Acerca del programa llamamos a la clase Acerca.java que explicaremos a continuación. Cuando pulsamos esta opción nos aparece una ventana en la que informamos de que se trata de un programa desarrollado para la asignatura Sistemas informáticos, así como el nombre de los miembros del grupo y nuestros directores de esta asignatura.

Cuando pulsamos la opción de Instrucciones de uso llamamos a la clase Ayuda2.java que nos dice como rellenar el documento de texto. Inicialmente nos dice que debemos de abrir dicho documento y en cada línea poner el nombre de la familia y a continuación un espacio y seguidamente separados por espacios los distintos grados de pertenencia de la familia que se ha colocado inicialmente a cada una de las características.

También nos indica que se pueden introducir tantas familias como se desee. Siempre y cuando después de rellenar las 206 características pulsemos un salto de línea para separarlas.

Nos menciona que es muy importante que se rellenen las 206 características ya que sino el programa no funcionara correctamente.

La ultima de las opciones es Rellenar características que llama a la clase Caracteristicas.java en la que nos dice como rellenar el documento Características.txt. Para cada una de las posiciones de la línea que comienza con el nombre de la familia nos indica cuales son las opciones que podemos poner en el documento.

Para los botones serán un numero que varia desde -1 hasta 3 y para los JComboBox las opciones que nos indique en dicha ventana.

En el texto que aparece en la ventana de características podemos ver entre paréntesis el nombre del panel en el que se encuentra la característica a rellenar, así como el nombre de dicha característica y las opciones que se tienen para rellenar dicha posición en la línea.

También podemos observar como el evento asignado a cerrar la herramienta con el aspa nos permite una ejecución más interactiva de la herramienta.

La clase PanelPrin.java es la que lleva todo el peso de la herramienta.

En ella se observa como en la constructora se lee del fichero Características.txt mediante la clase Ficheros.java. Entonces nos devuelve un ArrayList en el que en cada posición del array tenemos otro array cuya primera posición es el nombre de la familia y las posiciones posteriores cada uno de los grados de pertenencia de dicha familia en dicha posición que corresponde a una característica en concreto.

A continuación tenemos una nueva variable array familiasAct del tipo Tipo que lo que hace es tener dos atributos, uno con el nombre de la familia y otro con la suma de los grados de pertenencia que viene dado según las características, botones, que pulsemos. En el inicio todos los grados son cero y según se vayan pulsando botones el grado aumentara o simplemente si es -1 dicha familia desaparecerá del array.

Se pueden ver como se llama al método PonerPrin, que explicaremos a continuación.

Como algunos de los botones que se añaden a los paneles no deben de ser visibles hasta que otros de los que dependen estén pulsados, estos se ponen no visibles.

Vemos como se añaden todos y cada uno de los paneles. En cada uno de ellos se puede ver como es el titulo de las diferentes opciones que en el documento de Excel nos indico el biólogo con letras en negrita.

Para cada uno de los paneles se añaden tanto los botones de cada una de las características que el biólogo nos indico a continuación de las partes de las familias en negrita.

A parte de los botones las características que nosotras consideramos que se podían agrupar en un JComboBox como puede ser el numero de pétalos pues se ha puesto en el panel correspondiente.

También se ha puesto una etiqueta en la parte derecha de la herramienta con el nombre Familias y otras dos debajo de esta llamadas Aceptadas y Descartadas.

Debajo de cada una de ellas se ha puesto un Jlist que nos indicara las familias que están aceptadas debajo de la etiqueta que pone Aceptadas y las familias descartadas debajo de la etiqueta Descartadas.

Cada panel se inicializa con una imagen seguida del nombre del panel y dentro de dicho panel introducimos alguna imagen si corresponde además de inicializar todos los botones que el biólogo nos indico debajo de cada una de las partes de la familia en negrita.

Por ejemplo después de biotipo nos puso: Árbol, Arbusto, .... y esos son cada uno de los botones que inicializamos.

Para los botones que son condicionales, es decir que si no pulsamos una característica no deben de ser visibles para el usuario, solamente los creamos pero no estarán visibles hasta que no se pulse la característica de la que dependen para ser mostrados.

A continuación explicare cada uno de los eventos introducidos en dicha clase.

Para cada uno de los dos elementos Jlist introducidos en el panel tenemos un evento.

Para el Jlist que muestra las familias que hasta el momento están aceptadas. Por ello cada vez que se pulsa sobre alguna de las familias que están en dicho elemento obtenemos la aparición de una ventana con las características de dicha familia gracias a la clase Aceptados.java que es exactamente igual que la de anteriores herramientas.

Para el siguiente Jlist que muestra las familias que han sido descartadas hasta el momento también se pueden ver las características de ellas, solamente pulsando en ellas y esto llamara a la clase Aceptados.java que es la que realiza esta acción como en el anterior Jlist.

Para cada uno de los botones si el botón esta pulsado entonces llamamos al método Mirar con el array de las familias aceptadas actuales y con el numero del botón que hemos pulsado.

A continuación llamamos al método Poner que explicaremos a continuación.

Lo ultimo que ejecuta cada botón es deshabilitarlo para que ya no pueda ser pulsado de nuevo.

En caso de querer despulsarlo debemos de reiniciar la herramienta.

Como he comentado antes de realizar algún evento al pulsar el botón comprobamos que el botón esta habilitado ya que de esta forma evitamos que se puede pulsar el botón en



mas de una ocasión con la consecuencia de sumar de nuevo el grado de pertenencia de esa familia a la característica, en este caso botón correspondiente, pulsado.

En el caso de que algún botón active la visibilidad de otros botones basta con poner lo mismo que realizan el resto de los botones pero habilitando, es decir, poniendo como visibles los botones que dependen de él.

Para cada uno de los JComboBox debemos de conocer lo que esta seleccionado y a continuación si dicho elemento esta habilitado ejecutamos el método Mirar2 al que le pasamos las familias aceptadas actuales, el numero del botón y el elemento que esta seleccionado del JComboBox.

Después se llama al método Poner y deshabilitamos este elemento para que no pueda ser pulsado de nuevo.

En el método Mirar lo que hacemos es recorrer el array de todas las familias que siguen aceptadas hasta el momento obteniendo el nombre de la familia del array con el atributo nombreFamilia. Una vez hecho esto recorremos el array con todas las familias que es la variable familias que se obtiene de la lectura del documento de texto Características.txt cuyas posiciones tienen un array en el que en la primera posición esta el nombre de la familia y a continuación todas las características.

Cogemos el nombre de la familia es decir la posición cero de todos los arrays de cada posición del array principal, hasta que encontremos la coincidencia.

Cuando coinciden miramos la característica dentro del array de cada una de las posiciones del array principal. Como los botones los hemos inicializado desde el numero 1 al 206 y el array va desde el 0 no hay problemas en mirar la característica en dicho array ya que la primera posición como he mencionado anteriormente es el nombre de la familia.

Una vez encontrada la posición de la familia que estamos buscando cogemos ese valor mediante el método tratar característica lo obtenemos y dependiendo de ese valor hacemos una cosa u otra.

Si el valor es -1 la familia la ponemos en descartadas y el grado de la familia se pone a -1 directamente y pasa al array de descartados.

Si el valor es distinto de -1 cogemos el grado de la familia y le sumamos el valor obtenido del array.

Después como guardamos la suma en un ArrayList aux pues entonces debemos de crear una nueva variable de tipo Tipo para guardar tanto el nombre como el grado de pertenencia en dicha variable y añadirsele al ArrayList familiasAct en la posición en la que se encuentre el nombre de la familia y sobrescribirlo. Este array es un array en el que en cada posición tiene una variable de tipo Tipo

Para el método Mirar2 es exactamente igual pero lo que hacemos es comparar lo que tiene pulsado en el JComboBox y el texto que tiene en la posición indicada en el array mediante el método TratarCarac2.

Si el JComboBox tiene algún valor distinto del inicial, es decir, espacio vacío a la hora de reiniciar estos elementos debemos comparar con el espacio vacío que es con el valor que se inician y reinician dichos elementos.

Si el JComboBox tiene un espacio en blanco devolvemos un valor de cero para no sumar nada.

Después si esto no se cumple si el valor de la característica seleccionada coincide con la que hemos seleccionado devolvemos un valor de 3. Si el valor no coincide devolvemos un valor de -1.

Para el método PonerPrin que se realiza al comienzo de la ejecución de la herramienta, cogemos el nombre de la familia y el grado al principio no se muestra ya que no tiene, entonces el nombre lo ponemos en el Jlist, cuadro de texto que aparecen las familias aceptadas. Este método solo se ejecuta al comienzo de la herramienta.

Para el array descartados al principio se inicializa con el texto Ninguna descartada, ya que al comienzo de esta herramienta no existe ninguna familia descartada.

Para el método Poner que se ejecuta cada vez que pulsamos alguno de los botones o de los JComboBox lo que hacemos es anteriormente obtener el array familiasAct actualizado según haya o no familias en dicho array haremos una u otra cosa.

Si no hay elementos en dicha lista, en el Jlist aceptados mostraremos Ninguna aceptada y en el de descartados las familias que estén en el array descartados que serán las que tengan algún -1 en alguna de las posiciones correspondientes a los botones pulsados en la herramienta.

Si hay elementos en Jlist aceptados mostraremos el nombre de la familia seguido del grado de pertenencia que será la suma de todos los grados de pertenencia que están en las posiciones de los botones pulsados.

Si no hay familias descartadas en el Jlist correspondiente a descartados pondrá Ninguna descartada. Esto es lo que pasara en un primer momento de la ejecución de la herramienta.

En el método TratarCarac miramos si es igual a -1 en ese caso devolvemos -1 y si es otro numero lo pasamos a entero para que pueda ser sumado desde el método que este método es llamado.

En el método TratarCarac2 lo que hacemos es comparar con el espacio en blanco, si es así devolvemos cero. Si la característica a comparar es igual que la que tenemos en el documento de texto devolvemos un 3 y sino devolvemos un -1.

El método ordRapida ordena las familias de mayor a menor según el grado de pertenencia a esta. Es el numero que aparece a continuación del nombre de la familia.

En el método Reiniciar ponemos las listas de aceptados y descartados como en el metodo PonerPrin. En un primer momento todas las familias estarán aceptadas y en la lista de descartados pone Ninguna descartada.

Los botones se reinician, los botones se deseleccionan todos y los que no tienen que estar visibles no se ven.

Los JComboBox se reinician también a blanco.

La clase Tipo.java se crea un nuevo tipo con un atributo que contiene el nombre de la familia y otro con la suma de los grados de pertenencia. Al principio este grado es cero ya que al principio no se ha seleccionado ningún botón. El grado solo cambia cuando algún botón esta pulsado y se recalcula dicha suma.

La clase Ficheros.java es igual que en anteriores herramientas.

La clase Aceptados.java también es igual.

La clase Acerca.java es una nueva ventana en la que ponemos el nombre de la asignatura para la que hemos desarrollado las herramientas. En este caso Sistemas informáticos y el curso. En nuestro caso la asignatura es nuestro proyecto de fin de carrera. Además decimos los nombres de los miembros del grupo así como los dos directores de proyecto que hemos tenido.

Para cerrar dicha ventana podemos pulsar al botón Aceptar o pulsar el aspa, ya que pulsando el aspa el programa seguirá funcionando correctamente.

Es una novedad que añadimos a partir de la herramienta nítida en las demás herramientas ya que nos dimos cuenta de la facilidad de dejar sin funcionamiento la herramienta por una cosa tan sencilla como añadir un evento al pulsar dicho elemento, el aspa. Así conseguimos una herramienta más interactiva para el usuario.

La clase Ayuda2.java abre una nueva ventana en la que se muestra en el cuadro de texto lo leído en el documento de texto Ayuda2.txt mediante la clase Ficheros.java.

En dicho documento nos dice como rellenar el documento de texto Caracateristicas.txt mencionando que es imprescindible poner el nombre de la familia en cada línea, un espacio en blanco y seguidamente las 206 características unas con números y otras con texto separadas entre sí con espacios en blanco.

Después de cada introducción de las 206 debemos de pulsar intro, salto de línea para poder introducir nuevas familias.

Menciona como importantísimo introducir las 206 ya que sino el programa no funcionara.

La clase Características.java lee mediante la clase Ficheros.java el documento de texto Ayuda.txt y lo muestra en el cuadro de texto.

Para cerrar dicha ventana pulsamos el botón Aceptar o el aspa.

En el documento de texto Ayuda.txt nos dice cada una de las características que podemos añadir en el documento de texto Características.txt. Te indica en cada posición lo que puedes poner, estos números o texto se deben de introducir después del nombre de la familia.

Te dice los valores que se pueden introducir en cada una de las 206 posiciones a introducir.

Se comienza mencionando la posición, la característica a introducir y entre paréntesis el panel en el que se muestra. Después el rango de valores a introducir.

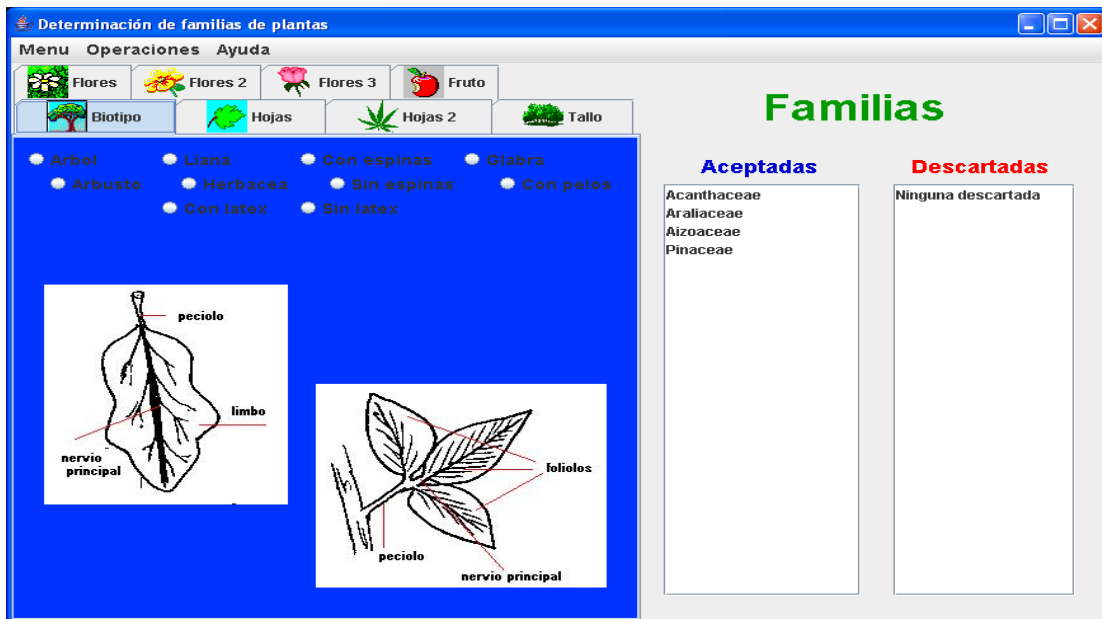
Si es un botón de -1 a 3 y en los JComboBox mencionamos todos los valores que podemos introducir tal y como ponemos en este documento. (Con sus respectivas mayúsculas y minúsculas).

Para finalizar comentar que todas las herramientas así como esta memoria se introducirán en una página web que se guardara en formato CD.

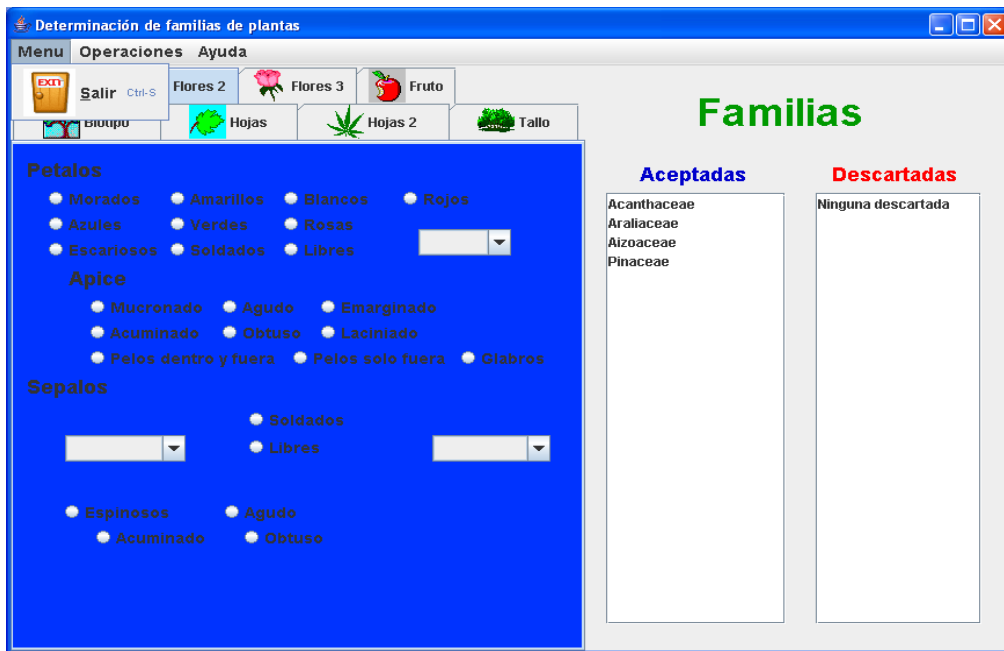
# EJECUCIÓN DE LA HERRAMIENTA

La pantalla inicial muestra todas las familias en la columna de Aceptadas y en Descartadas, pone Ninguna descartada.

Podemos observar como todos los botones estan desactivados y como se ven las distintas opciones del menu asi como todas las pestañas que nos indico el biólogo en negrita en el documento de Excel que nos pasó. En cada una de las pestañas se muestran los distintos botones que se pueden pulsar asi como los JComboBox que hemos introducido con otras opciones que nos indicó.

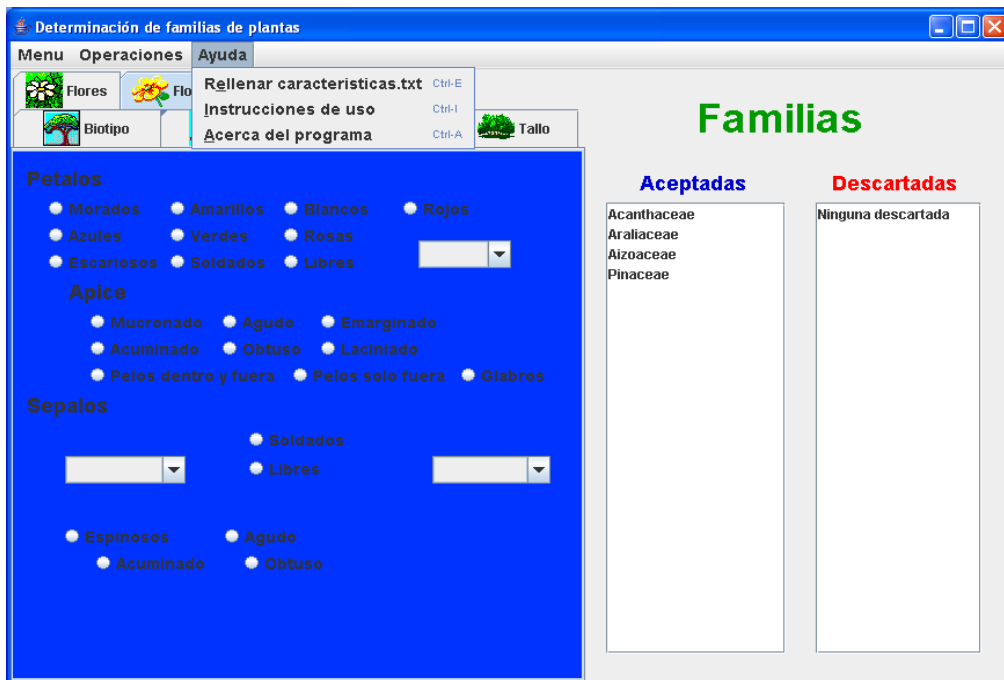


En la siguiente imagen vemos como se muestra la opción del menú Menu Salir. También se puede ver las diferentes opciones que tienes en la pestaña Flores 2, que aparecen tanto botones como JComboBox.



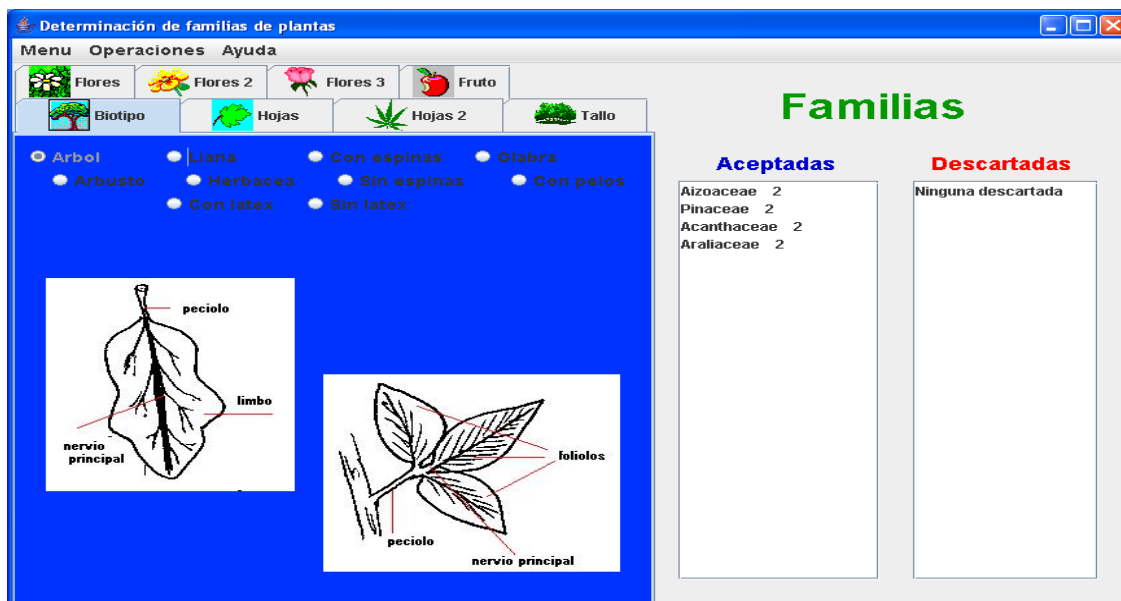
Para el menú de opciones ayuda vemos como se muestran todas las opciones que hemos comentado anteriormente.

También se puede acceder a estas opciones al igual que a la de Salir de la anterior imagen con las opciones rápidas. Es decir con ctrl. + una letra. Esto se puede ver en la parte derecha de cada opción.



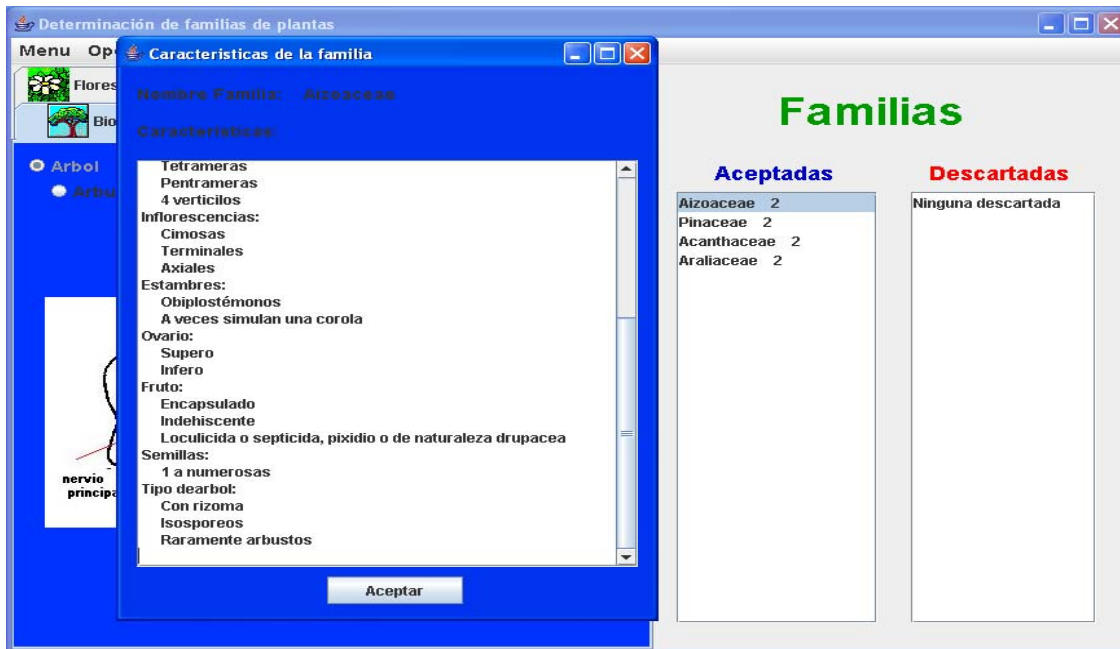
Pulsando en cualquier boton vemos como cambian los grados de las diferentes familias y se produce en este caso la suma de lo que dichas familias tienen en la posición de la característica grado con lo que poseían en un principio, osea, con cero.

En este caso todas las familias tienen grado 2 en dicha característica y no aparece ninguna familia descartada ya que en dicha posición no tenían ninguna un grado de pertenencia igual a -1.

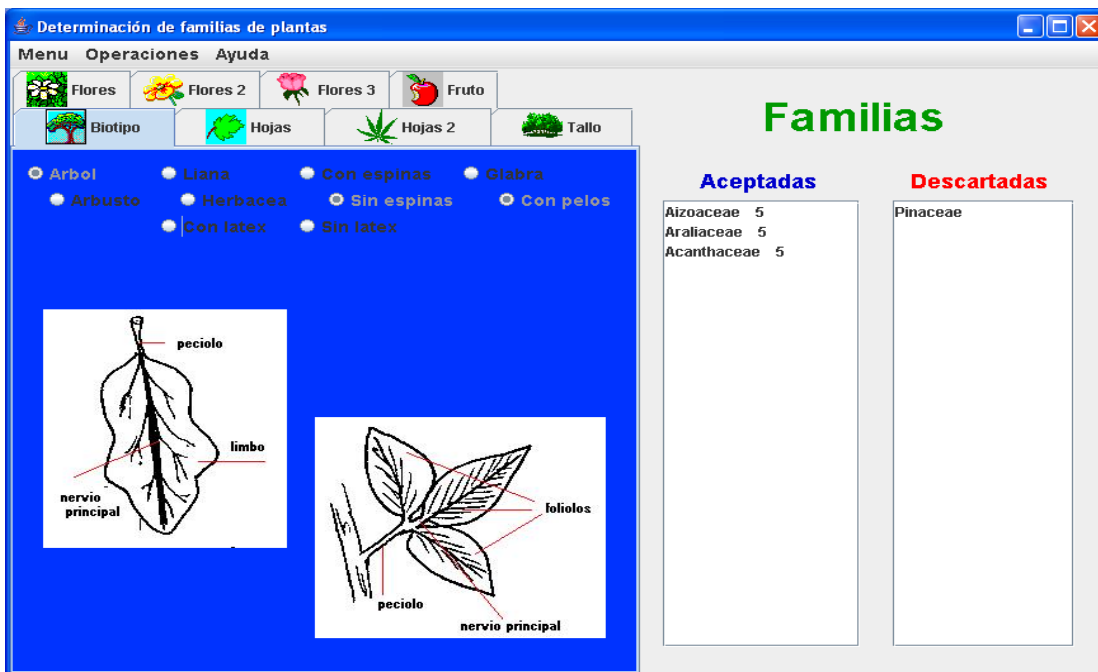


Como en otras herramientas si pulsar sobre la familia tanto en el cuadro que muestra las familias aceptadas como las descartadas podemos ver que se muestra una ventana con el nombre de la familia pulsada y sus características, están como siempre se leen del fichero plantas2.txt.

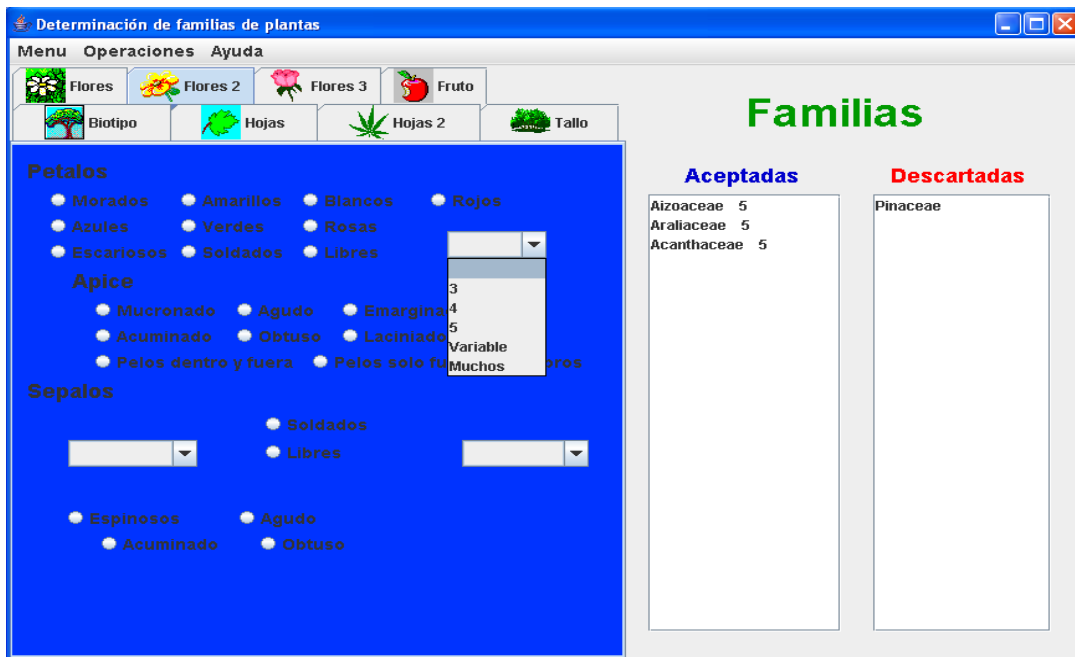




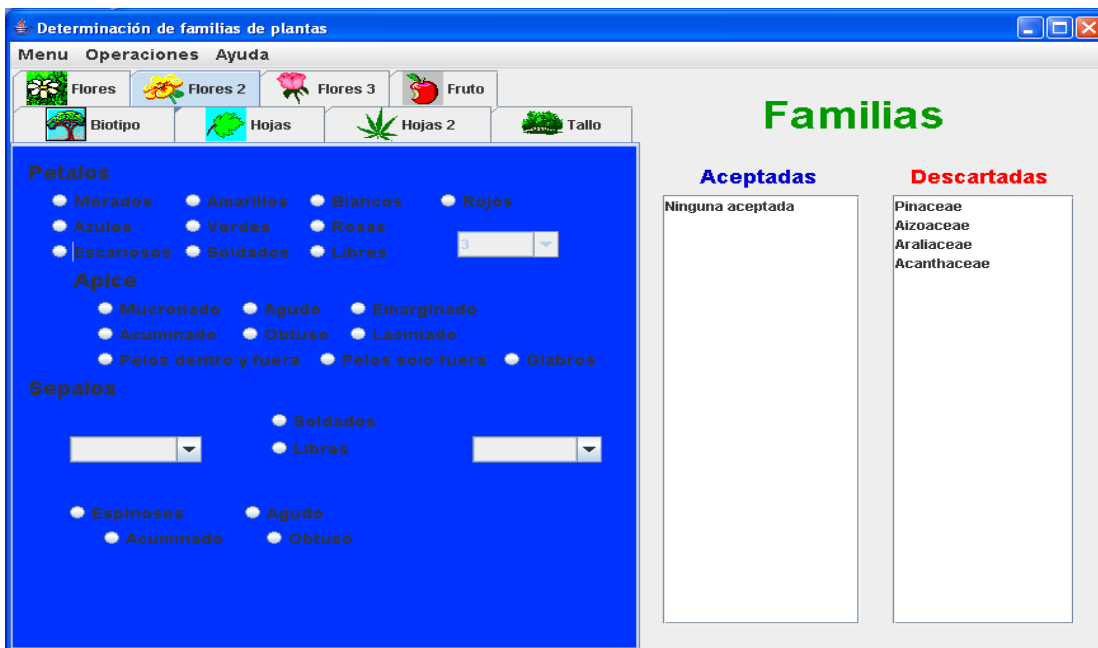
Podemos ver como aumenta el grado de las familias hasta sumar cinco. También podemos ver como la familia *Pinaceae* se muestra en las descartadas, eso quiere decir que en alguna de las características pulsadas tenia un -1 como grado de pertenencia.



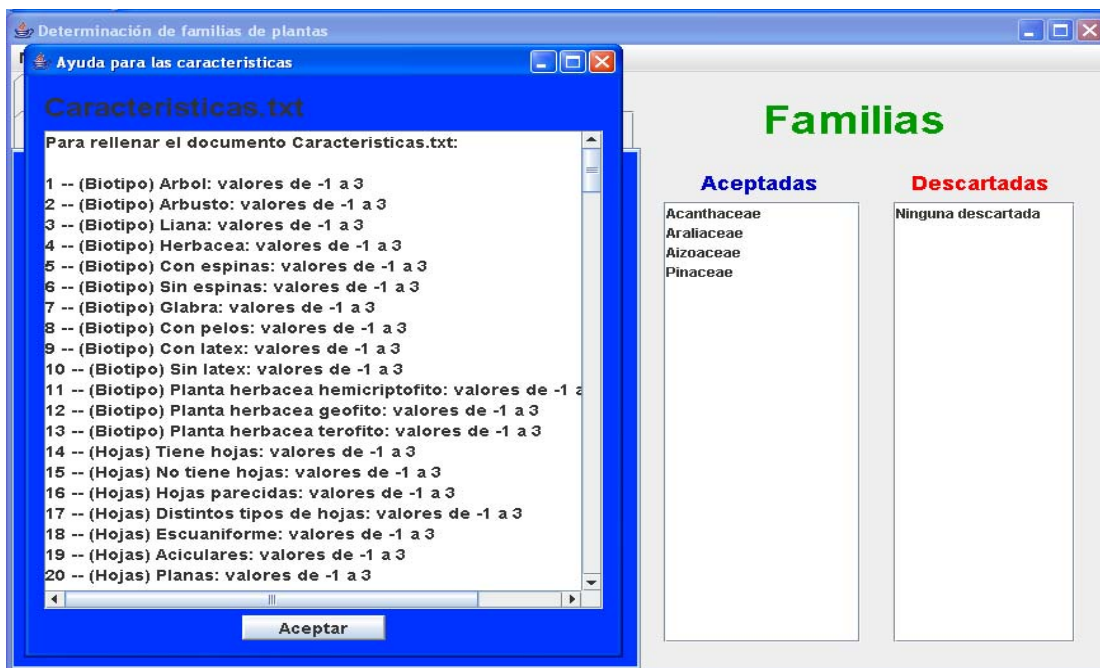
En esta otra imagen vemos las opciones que tiene el JComboBox número de pétalos.



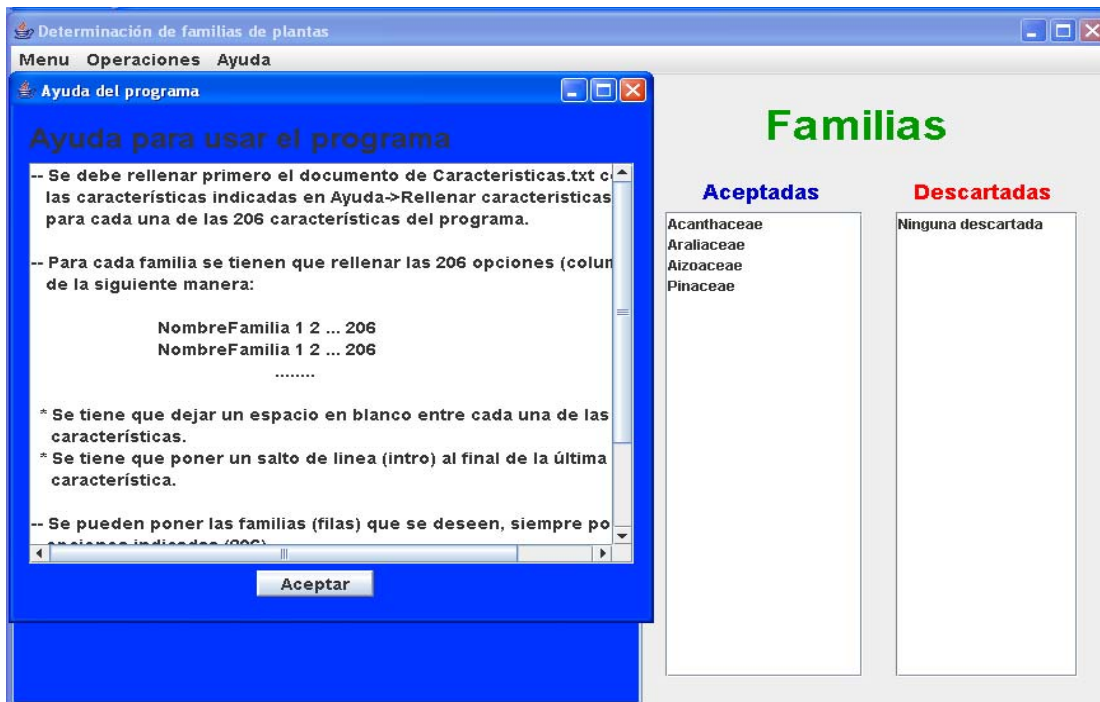
Para la opciones de tener 3 en la característica numero de pétalos no la posee ninguna por lo tanto todas pasan a descartadas. El paso siguiente sería reiniciar la herramienta.



En esta imagen se puede ver lo que se muestra cuando pulsamos en el menu de Ayuda Rellenar características.



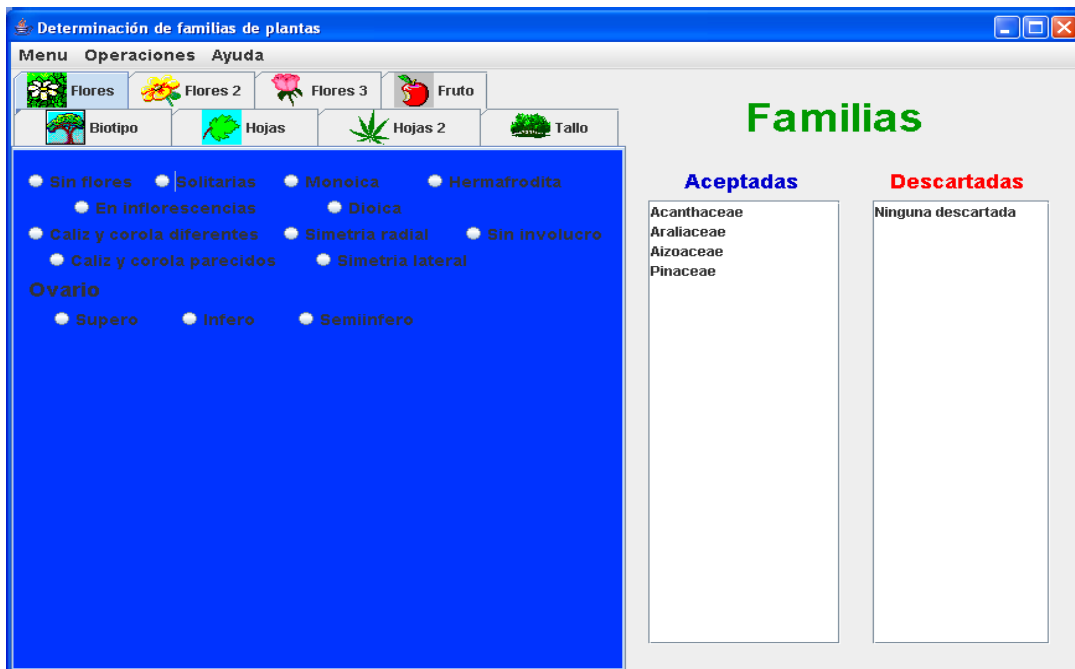
En esta imagen podemos observar lo que se muestra pulsando en el menú de Ayuda Instrucciones de uso.



Para la opción Acerca del programa, pulsando sobre ella o pulsando ctrl. + A nos sale la siguiente ventana.



Ahora vamos a mostrar las opciones no visibles al usuario, bueno una de ellas cuando pulsamos el boton Sin Flores. Así se muestra al principio.



Y así se muestra después de pulsar la opción Sin Flores. Se muestran el resto de los botones que dependían de él.

**Determinación de familias de plantas**

Menu Operaciones Ayuda

Flores
 Flores 2
 Flores 3
 Fruto

Biotipo
 Hojas
 Hojas 2
 Tallo

# Familias

☒ Sin flores
 ☐ Solitarias
 ☐ Monoica
 ☐ Hermafrodita
 ☐ En inflorescencias
 ☐ Dioica

☐ Caliz y corola diferentes
 ☐ Simetría radial
 ☐ Sin involucre
 ☐ Caliz y corola parecidos
 ☐ Simetría lateral

**Ovario**
☐ Supero
 ☐ Infero
 ☐ Semiinfero

☐ Con esporangios
 ☐ Isosporeos
 ☐ Sin esporangios
 ☐ Heterosporeos

**Aceptadas**

Aizoaceae 0  
Pinaceae 0  
Acanthaceae 0  
Araliaceae 0

**Descartadas**

Ninguna descartada

## APÉNDICE: HERRAMIENTA XFUZZY

La herramienta usada para la inferencia borrosa será el programa llamado XFuzzy, nosotras hemos utilizado el que se puede descargar en la siguiente página:

[http://www.imse.cnm.es/Xfuzzy/index\\_sp.html](http://www.imse.cnm.es/Xfuzzy/index_sp.html)

En dicha pagina web también se puede descargar un manual en español que explica el uso de la herramienta, aunque este manual es de gran ayuda, nosotras proporcionaremos el siguiente manual que pensamos que será de gran utilidad para cualquier persona que quiera realizar alguna tarea con la herramienta ya que como nosotras puede que no tenga conocimiento previo de esta y entonces al nosotras encontrarnos en misma situación, pues ira conociendo paso a paso como hay que realizar un sistema en esta herramienta.

Inicialmente tenemos que descargarnos la versión de XFuzzy que se desee pero conviene descargarse siempre la última versión debido a la corrección de los posibles problemas que pudiesen tener versiones anteriores.

El manual sirve para la herramienta XFuzzy en versión 3.0. Para ello el ordenador se va a descargar esta herramienta debe de tener instalado el Java Runtime Enviroment (JRE). En caso de no disponer de este en su ordenador, se proporciona un enlace en la pagina web menciona anteriormente para poder descargarte cualquier versión de JRE aunque siempre es conveniente descargarse la ultima versión, como he dicho antes.

Una vez que tienes instalado el JRE en su ordenador, procedemos a la descarga del XFuzzy versión 3.0. Te tienes que ir al sitio de tu ordenador donde has querido que se descargara la herramienta.

Después pinchas sobre este haciendo doble clic y te saldrá una pantalla como esta:



En ella debes de poner el directorio donde quieres que se instale la herramienta. Se este directorio no existe el proceso de instalación lo creara. Debes de pulsar en el botón que pone Browse para poder introducir un directorio.

Elija también el directorio donde se encuentra el ejecutable JRE que previamente se ha descargado o si ya lo tenía instalado en su ordenador, vaya a dicho directorio. Normalmente es el subdirectorio bin de la instalación del J2SE.

Debe de pulsar sobre el botón Browse que se encuentra a la derecha de la etiqueta Path for java ejecutables

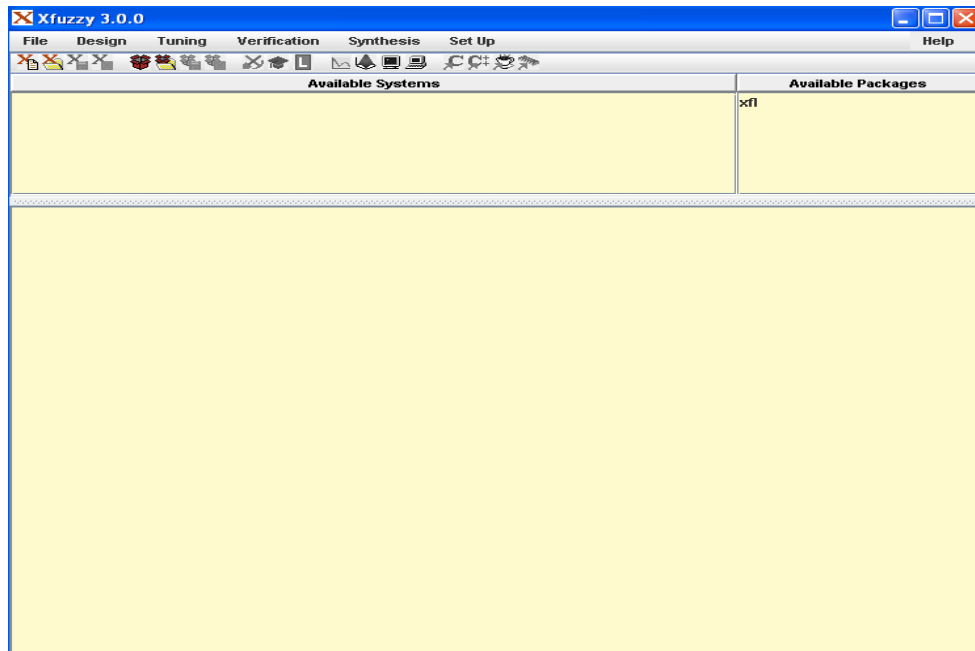
Además debe de elegir el navegador para mostrar los ficheros de ayuda. Dando al botón Browse de la derecha de la etiqueta HTML Viewer para poder decir el directorio donde se encuentra el navegador.

Una vez realizado esto debemos de dar al botón install y entonces nos instalara la herramienta XFuzzy en el directorio que le hayamos señalado para que la instale, es decir, en la primera línea de esta ventana que nos aparece.

Cuando haya llegado al 100% la herramienta ya esta totalmente lista para ser ejecutada.

Debemos de ir a la carpeta donde hemos instalado la herramienta y acceder a la carpeta bin que es donde se encuentran los ejecutables del XFuzzy.

Entonces pulsamos doble clic en el ejecutable que tiene por nombre xfuzzy y entonces nos aparecerá una ventana como esta:

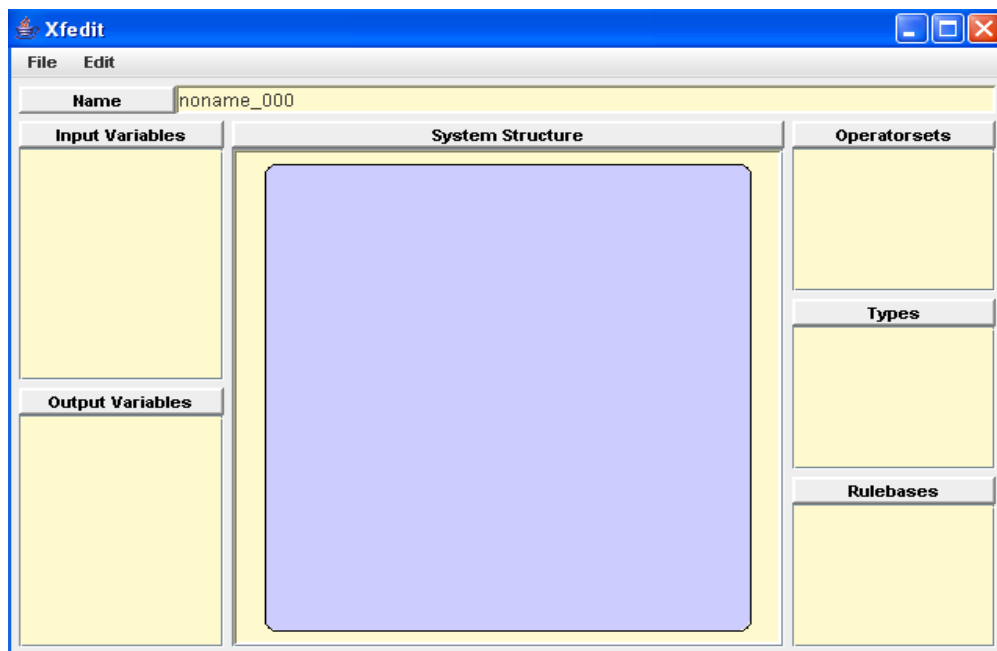


Entonces para realizar un nuevo sistema borroso debemos de pulsar en el menú File y elegir la opción New System.

El nuevo sistema tiene el nombre noname\_000 que es el que el programa le asigna por defecto.

Para poder cambiarlo debemos de pulsar en el nombre del sistema y elegir la opción Edit System del menú Design.

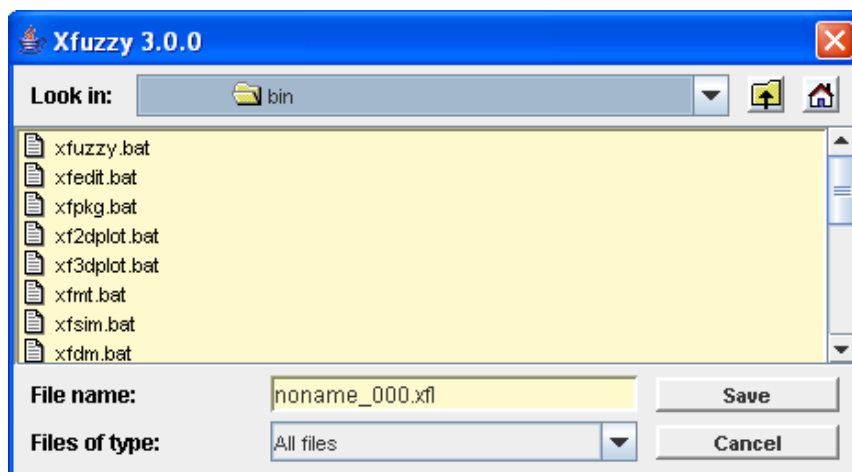
Entonces nos sale una ventana como la siguiente:





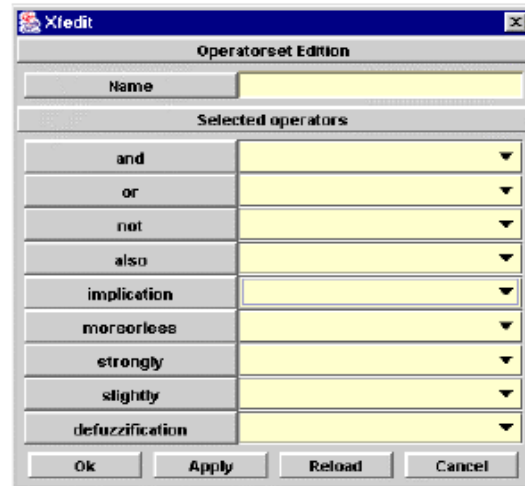
Y lo que debemos de hacer es pulsar el menú File y elegir la opción SaveAs y entonces al aparecernos esta ventana escribimos el nombre que queremos darle a nuestro sistema, además del directorio donde queremos que nos lo guarde. Aunque es conveniente que el directorio no se cambie ya que podría siempre es mejor dejar lo que las herramientas guardan en el directorio que ellas te proporcionan por defecto.

Una cosa muy importante la extensión que nos pone .xfl debemos de dejársela ya que es lenguaje que usa esta herramienta.



Cuando le hayamos dado el nombre al sistema y le hallamos dejado la extensión .xfl simplemente debemos de pulsar en botón Save y entonces volveremos a la pantalla anterior viendo que el donde pone Name nos aparece el nombre que le hallamos dado a nuestro sistema.

Para comenzar debemos de crear nuevos operadores, para ello pulsamos en el menú Edit y pulsamos la opción New Operatorset.



Lo primero que debemos de realizar es darle un nombre al operador usado.

Después debemos de elegir los operadores. Los operadores pueden constar de varias funciones. El programa proporciona funciones binarias por defecto que se muestran en la pantalla siguiente, y funciones unarias que se muestran a continuación de las anteriores:

Funciones binarias

Nombre	Tipo	Descripción Java
min	T-norm	$(a < b ? a : b)$
prod	T-norm	$(a * b)$
bounded_prod	T-norm	$(a + b - 1 > 0 ? a + b - 1 : 0)$
drastic_prod	T-norm	$(a == 1 ? b : (b == 1 ? a : 0))$
max	S-norm	$(a > b ? a : b)$
sum	S-norm	$(a + b - a * b)$
bounded_sum	S-norm	$(a + b < 1 ? a + b : 1)$
drastic_sum	S-norm	$(a == 0 ? b : (b == 0 ? a : 0))$
dienes_resher	Implication	$(b > 1 - a ? b : 1 - a)$
mizumoto	Implication	$(1 - a + a * b)$
lukasiewicz	Implication	$(b < a ? 1 - a + b : 1)$
dubois_prade	Implication	$(b == 0 ? 1 - a : (a == 1 ? b : 1))$
zadeh	Implication	$(a < 0.5    1 - a > b ? 1 - a : (a < b ? a : b))$
goguen	Implication	$(a < b ? 1 : b / a)$
godel	Implication	$(a <= b ? 1 : b)$
sharp	Implication	$(a <= b ? 1 : 0)$

## Funciones unarias

Nombre	Parámetro	Descripción Java
not	-	$(1-a)$
sugeno	l	$(1-a)/(1+a*l)$
yager	w	$\text{Math.pow}( ( 1 - \text{Math.pow}(a,w) ) , 1/w )$
pow	w	$\text{Math.pow}(a,w)$
parabola	-	$4*a*(1-a)$

Aquí se muestran los métodos de desfuzzificación que están definidos en el paquete estándar:

Nombre	Tipo	Definido para
CenterOfArea	Conventional	any function
FirstOfMaxima	Conventional	any function
LastOfMaxima	Conventional	any function
MeanOfMaxima	Conventional	any function
FuzzyMean	Simplified	triangle, isosceles, trapezoid, bell, rectangle, singleton
WeightedFuzzyMean	Simplified	triangle, isosceles, trapezoid, bell, rectangle
Quality	Simplified	triangle, isosceles, trapezoid, bell, rectangle
GammaQuality	Simplified	triangle, isosceles, trapezoid, bell, rectangle
MaxLabel	Simplified	singleton
TakagiSugeno	Simplified	parametric

Estas funciones por defecto se encuentran en el paquete estándar xfl, que coincide con la extensión de nuestro sistema. Este lenguaje de especificación en el caso de XFuzzy 3.0 es XFL3 permite al usuario definir sus propias funciones y por lo tanto sus propios operadores.

Para definir tus propias funciones binarias basta con leer el apartado del manual de la pagina web en el apartado que pone Definición de funciones binarias. Para definir tus propias funciones unarias hay que buscarlo en el apartado Definición de funciones unarias.

En este manual lo explica de manera sencilla para que el usuario puede entender como realizar esta tarea ya que una novedad de esta herramienta es la posibilidad que se le

proporciona al usuario de definir sus propias funciones binarias, unarias, sus propios operadores, además de sus propias funciones de pertenencia y métodos de desfuzzificación.

Conociendo estas funciones binarias y las funciones unarias podemos crear un operador de nuestro sistema. Para ello es importante conocer lo que cada función puede producir, además de conocer lo que estas pueden proporcionar como resultado cuando combinamos varias de ellas.

También tenemos la posibilidad de crear funciones, métodos de defuzzificación, etc. En el manual que se encuentra en la página web se puede ver como crear las funciones y los métodos de una manera sencilla.

Para conocer mejor como se muestran las funciones añadidas al archivo xfl de nuestro sistema vamos a poner el formato de los operadores.

```
operatorset identifier {
    operator assigned_function(parameter_list);
    operator assigned_function(parameter_list);
    ..... }
```

Operatorset sirve para conocer que es un operador. Identifier sirve para conocer el nombre que le hemos dado nosotros al operador.

Entonces después de la etiqueta que pone operator le sigue la función binaria o unaria que hemos introducido en la herramienta seguida de la lista de parámetros.

No es necesario especificar todos los operadores ya que si uno no está definido asumimos su valor por defecto. En la siguiente imagen se pueden ver los valores por defecto de las distintas funciones tanto de las binarias como de las unarias.

Operador	Tipo	Función por defecto
and	binary	$\min(a,b)$
or	binary	$\max(a,b)$
implication, imp	binary	$\min(a,b)$
also	binary	$\max(a,b)$
not	unary	$(1-a)$
very, strongly	unary	$a^2$
moreorless	unary	$(a)^{(1/2)}$
slightly	unary	$4*a*(1-a)$
defuzzification, defuz	defuzzification	center of area

La siguiente imagen muestra un operador llamado systemop que tiene las funciones and, or imp, strongly y moreorless. Se puede ver como delante de cada una de lo que se introduce en cada función pone xfl eso quiere decir que la función que viene a continuación se coge del paquete xfl.

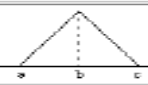
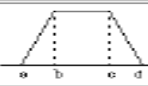
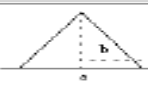

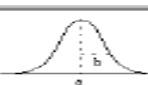
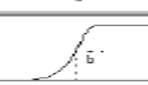


```
operatorset systemop {
  and xfl.min();
  or xfl.max();
  imp xfl.min();
  strongly xfl.pow(3);
  moreorless xfl.pow(0.4);
}
```

Por ejemplo:

xfl.min() la función mínimo se coge del paquete estándar xfl.

Una vez hecho esto lo que primeramente tenemos que pensar es cuantos tipos de datos vamos a tener y dependiendo de cuales van a ser las variables de entrada al sistema, cuales van a ser las variables de salida comenzar creando nuevos tipos.

Para ello debemos de conocer las diferentes funciones de pertenencia que podemos tener. En la siguiente imagen se pueden mostrar las diferentes funciones de pertenencia que podemos tener como defecto.

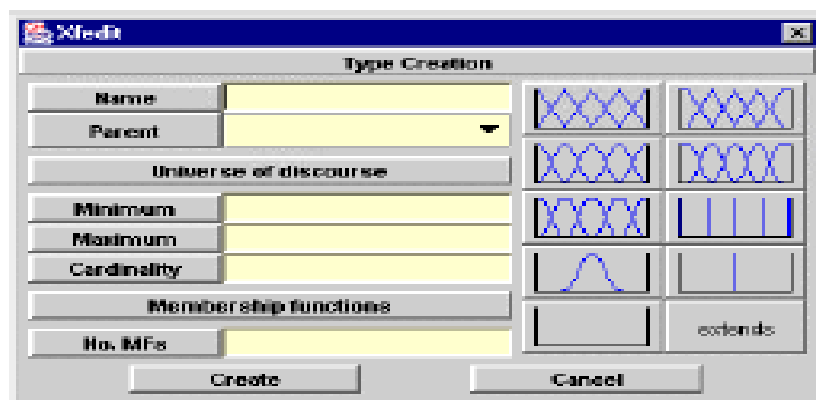
Nombre	Parámetros	Descripción
triangle	a,b,c	
trapezoid	a,b,c,d	
isosceles	a,b	
slope	a,m	
bell	a,b	
sigma	a,b	
rectangle	a,b	
singleton	a	
parametric	unlimited	-

Esta herramienta también nos proporciona la posibilidad de crear nuestros propias funciones de pertenencia, si se desea realizar esto en el manual de la pagina web se puede encontrar la manera de hacerlo. Esto se puede encontrar en bajo el apartado Definición de funciones de pertenencia. El manual lo explica de manera clara como realizar esto.

De todas formas con las funciones de pertenencia que podemos encontrar en la herramienta por defecto, seria suficiente para iniciarse en el mundo de la lógica borrosa o inferencia borrosa.

Una vez pensados los tipos que va a usar su sistema debemos de elegir la opción New Type del menú Edit.

Entonces nos aparecerá una ventana como esta:



En ella elegimos el nombre de nuestra función de pertenencia y a continuación como la opción de Parent esta desactivada, debemos de pulsar en las opciones que están por defecto en la parte derecha de la imagen. Aquí están representadas todas las funciones de pertenencia que existen en el paquete xfl que esta por defecto en la herramienta.

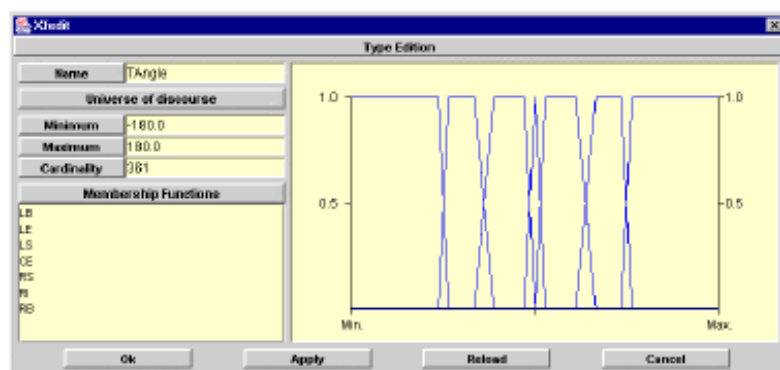
La opción Parent solo esta desactivada para cuando realizamos el primer tipo pero en los sucesivos como la herramienta permite la herencia de tipos podemos extender de cualquier tipo anteriormente creado pulsando sobre la opción Parent o simplemente en la pantalla de la derecha nos aparece la opción de extends y elegimos entre los tipos previamente creados para realizar su extensión. Es decir, esta herramienta permite herencia de tipos.

Después de haber hecho esto debemos de elegir el máximo, el mínimo y la cardinalidad de este tipo. El mínimo significa el mínimo valor que puede tomar este tipo, el máximo por tanto el máximo valor que puede tomar el tipo y por ultimo la cardinalidad que es el numero de elemento discretos. Si la cardinalidad no se especifica se asume que por defecto es 256.

Cuando no se definen los límites del universo en discurso se asume que por defecto son de 0 a 1.

A continuación se debe de especificar el número de funciones de pertenencia de la partición.

Después de esto solo resta pulsar el botón Create y entonces nos aparecerá una ventana como esta.



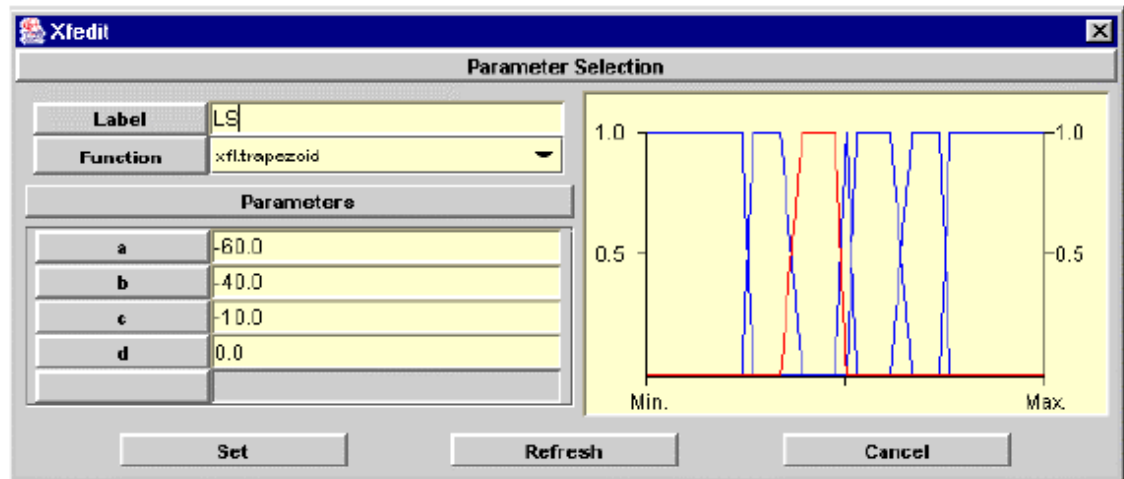
Una vez creado el tipo podemos editarlo mediante esta ventana. Podemos cambiar el nombre del tipo así como los valores del universo de discurso, es decir, los límites de este y su cardinalidad.

Como la modificación en la definición del universo de discurso puede afectar a las funciones de pertenencia se realiza una validación de los parámetros de las funciones de pertenencia antes de salvar las modificaciones apareciendo un mensaje de error cuando la definición de una función de pertenencia es inválida una vez introducidas las modificaciones.

También podemos cambiar los valores que da por defecto a las funciones de pertenencia de la partición que crea por defecto. El nombre de esta es mf0 para la primera función, mf1 para la segunda y así para todo el número de funciones de pertenencia de la

partición que se hayan puesto en la pantalla de creación de un nuevo tipo.

Lo anterior se puede conseguir haciendo doble clic sobre la función que queramos modificar. Entonces nos aparecerá una ventana como la siguiente:



Entonces en ella podremos cambiar el nombre de la función de pertenencia sobre la que hemos pulsado en la pantalla anterior.

También podemos cambiar la clase de función de pertenencia aunque esto al ser elegido para todas las funciones de pertenencia de la partición puede que no sea un cambio óptimo a realizar en la modificación de esa función.

Lo importante será cambiar tanto el nombre que el sistema nos da por defecto como introducir los valores de los parámetros de la función.

En la pantalla que hemos visto anteriormente donde muestra todas las funciones de pertenencia del paquete xfl se pueden ver los parámetros a que corresponden en la función y así conocer lo más conveniente para la creación de nuestro tipo.

Entonces pulsamos refresh para ver las modificaciones que hemos realizado sobre la función, estas se pueden ver en la grafica que aparece a la derecha en esta pantalla mostrándose en rojo la función de pertenencia que estamos modificando.

Ahora pulsamos el botón Set y volvemos para realizar lo explicado en todas las funciones de pertenencia de la partición.

Tenemos que realizar el mismo proceso para cada uno de los tipos que queremos introducir en nuestro sistema.



Para entender mejor el código del archivo de salida que nos proporciona el sistema mostraremos el formato de los tipos:

```
type identifier [min, max; card] {  
    label membership_function(parameter_list);  
    label membership_function(parameter_list);  
    ..... }  
}
```

Type por la definición de tipo. Identifier por el nombre del tipo que le vamos a dar. Entre corchetes vemos que se pone min, max y card estos serán los valores que hayamos introducido en el universo de discurso al crear el tipo.

Después vemos en las siguientes líneas del cuerpo de esta definición del tipo una etiqueta label la que se va a sustituir por todos los nombres de cada una de las funciones de pertenencia de la partición. Bajo la etiqueta membership\_function van a aparecer el nombre de las funciones de pertenencia en nuestro caso las del paquete estándar que irán precedidas de xfl. y a continuación el nombre de la función de pertenencia entre las que el paquete xfl nos proporciona.

Después la etiqueta parameter\_list se encontraran los valores de los parámetros que le hayamos establecido a cada una de las funciones de pertenencia de la partición.

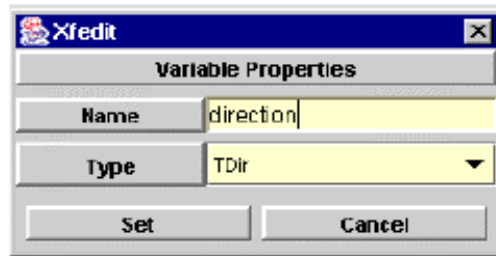
Un ejemplo puede ser el siguiente:

```
type Tinput1 [-90,90] {  
    NM xfl.trapezoid(-100,-90,-40,-30);  
    NP xfl.trapezoid(-40,-30,-10,0);  
    CE xfl.triangle(-10,0,10);  
    PP xfl.trapezoid(0,10,30,40);  
    PM xfl.trapezoid(30,40,90,100);  
}
```

A continuación debemos de introducir las variables de entrada al sistema como las variables de salida del sistema.

Para crear las variables de entrada solo debemos de pulsar sobre la opción del menú Edit, New input variable.

Entonces nos saldrá una pantalla como la siguiente:

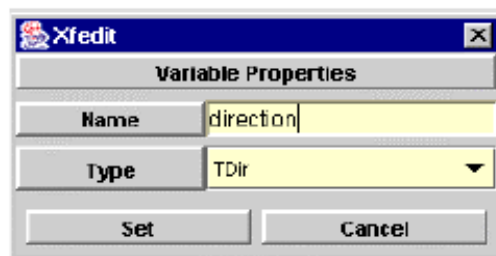


Entonces introducimos el nombre de la variable de entrada, normalmente se pone in seguido del nombre de la variable de entrada que querías introducir debido a que luego en el código será mucho más comprensible que las variables de entrada siempre van precedidas de in. Es una forma muy eficiente de identificarlas.

Después elegimos el tipo que queremos que tenga esta variable, entre los tipos que anteriormente hemos introducido y pulsamos en el botón Set.

Esto lo debemos de realizar para todas las variables de entrada que queramos introducir en nuestro sistema.

Para las variables de salida se debe de pulsar en el menú Edit, New output variable y aparece una ventana como esta:



Elegimos el nombre y el tipo. Al igual que en las variables de entrada precedíamos el nombre con in para las de salida va precedido por out para la mejor identificación de las variables de salida en el archivo que crea nuestro sistema.

Pulsamos el botón Set y así para todas las variables de salida.

Normalmente el número de variables de salida coincide con el número de operadores que hayamos introducido previamente como luego explicaremos a continuación en las reglas.

Lo siguiente y último que debemos de editar en nuestro sistema es la base de reglas.

Entonces pulsamos sobre la opción Edit, New rulebase y entonces nos aparecerá una ventana como esta.

Rule	Weight	Premise	Conclusion
0	1.0	if y = far )	dir = backward
1	1.0	if y == far & oldir == stop )	dir = backward
2	1.0	if y == far & oldir > stop )	dir = forward
3	1.0	if y == near S x != CE S angle == LE S angle == RI )	dir = forward
4	1.0	if y == near S x != CE S ( angle < LE   angle > RI ) )	dir = backward
5	1.0	if x == CE S y == near S angle == LS S angle == RS )	dir = backward
6	1.0	if x == CE S y == near S ( angle < LE   angle > RI ) )	dir = backward
7	1.0	if x == CE S y == near S angle > CE S angle < RI )	dir = forward
8	1.0	if x == CE S y == near S angle < CE S angle > LE )	dir = forward
9	1.0	if x == CE S y < near S ( angle < LE   angle > RI ) )	dir = backward
10	1.0	if x == CE S y < near S angle > CE S angle < RI )	dir = forward
11	1.0	if x == CE S y < near S angle < CE S angle > LE )	dir = forward
12	1.0	if x == CE S y < near S angle == CE )	dir = stop
*			

En la parte izquierda vemos como se puede introducir el nombre de la base de reglas, también se puede elegir el nombre del operador usado y se muestran las variables de entrada y de salida del sistema. Por eso previo a esta edición debemos de haber introducido tanto el operador que queramos usar como las variables de entrada y salida del sistema y los tipos de estas.

En la parte derecha se muestra el contenido de la base de reglas. Este contenido se muestra en forma libre, tabular o matricial.

En el formato libre se ven tres columnas, una para el peso de la regla, la segunda columna muestra el antecedente de la regla y la tercera y última columna muestra el consecuente.

La segunda columna muestra un campo editable en el que se muestra el símbolo ? que indica un termino vacío.

La tercera columna se puede editar pulsando el botón -> que se encuentra en la parte inferior de la pantalla.










Además se pueden introducir nuevas reglas introduciendo valores en la ultima fila que se encuentra marcada por el símbolo \*

En la parte de abajo se muestran los botones con todas las opciones que podemos incluir en las reglas.

Podemos unir los términos creados con conjunciones (&), con disyunciones (|). Podemos modificar los términos con not (!), con more or less (~), con slightly (%), con strongly (+). también podemos modificar los términos que relacionan una variable con una de las funciones de pertenencia de la partición a la que pertenece el tipo de esa variable con las cláusulas equal (==), not equal (!=), etc...

Todo esto se muestra en la siguiente imagen:

Términos que relacionan variable con una de las funciones de pertenencia de la partición.

Proposiciones básicas	Descripción	Representación
variable == label	equal to	
variable >= label	equal or greater than	
variable <= label	equal or smaller than	
variable > label	greater than	
variable < label	smaller than	
variable != label	not equal to	
variable %= label	slightly equal to	
variable ~= label	moreorless equal to	
variable += label	strongly equal to	

Términos complejos

Proposiciones complejas	Descripción
proposition & proposition	and operator
proposition   proposition	or operator
!proposition	not operator
%proposition	slightly operator
~proposition	moreorless operator
+proposition	strongly operator

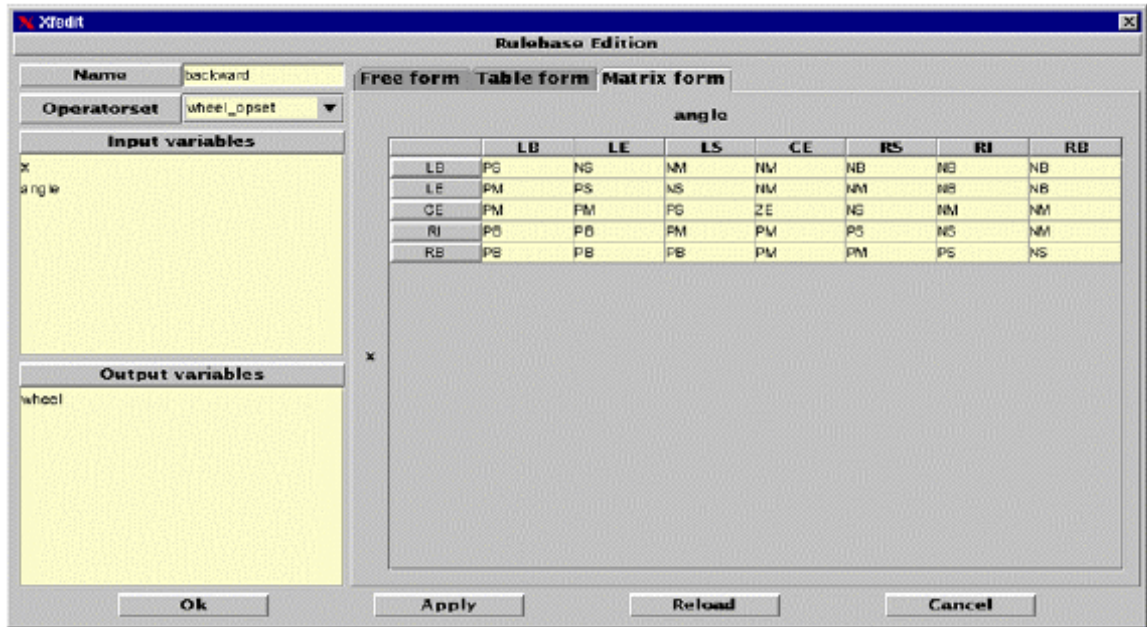
Un ejemplo de una base de reglas será:

```
rulebase base1(input1 x, input2 y : output z) using systemop {
  if( x == medium & y == medium) -> z = tall;
  [0.8] if( x <=short | y != very_tall ) -> z = short;
  if( +(x > tall) & (y ~= medium) ) -> z = tall;
  ..... }
```

En la parte de abajo se muestra el nombre de la variable así como la función de pertenencia de la partición con la que se relaciona.

Rule	if	then	output
0	1.0	if	dd == stop -> wheel = ZE dir = stop
1	1.0	if	dd == backward -> dir = backward
2	1.0	if	dd == forward -> dir = forward
3	1.0	if	bw == NB & dd == backward -> wheel = NB
4	1.0	if	bw == NM & dd == backward -> wheel = NM
5	1.0	if	bw == NS & dd == backward -> wheel = NS
6	1.0	if	bw == ZE & dd == backward -> wheel = ZE
7	1.0	if	bw == PS & dd == backward -> wheel = PS
8	1.0	if	bw == PM & dd == backward -> wheel = PM
9	1.0	if	bw == PB & dd == backward -> wheel = PB
10	1.0	if	fw == NB & dd == forward -> wheel = NB
11	1.0	if	fw == NM & dd == forward -> wheel = NM
12	1.0	if	fw == NS & dd == forward -> wheel = NS
13	1.0	if	fw == ZE & dd == forward -> wheel = ZE
14	1.0	if	fw == PS & dd == forward -> wheel = PS
15	1.0	if	fw == PM & dd == forward -> wheel = PM
16	1.0	if	fw == PB & dd == forward -> wheel = PB

Para cuando se muestra el contenido de las reglas en formato tabular es mejor esta vista para introducir los operadores and y equal. Cada regla dispone de un campo para introducir una variable de entrada y una lista desplegable para introducir el valor de la función de pertenencia de la partición. No es necesario introducir todos los campos de la regla ya que habrá un momento en el que alguna de las variables de entrada no nos interese su valor en la regla que estamos construyendo.



El formato matricial esta diseñado para describir una base de reglas con dos entradas y una salida. Por ejemplo: if (x==X & y==Y) -> z=Z. Las reglas con factor de confianza 1.0 formadas por la conjunción de dos igualdades. Para otra base de reglas no se podrá mostrar esta vista.

Para que el usuario entienda mejor el archivo de salida del sistema ponemos aquí el formato de las reglas:

```
rulebase identifier (input_list : output_list) using operatorset {
    [factor] if (antecedent) -> consequent_list;
    [factor] if (antecedent) -> consequent_list;
    ..... }
```

Después de la etiqueta rulebase va el nombre de la base de reglas. Entre paréntesis se muestran la lista de variables de entrada y las lista de variables de salida.

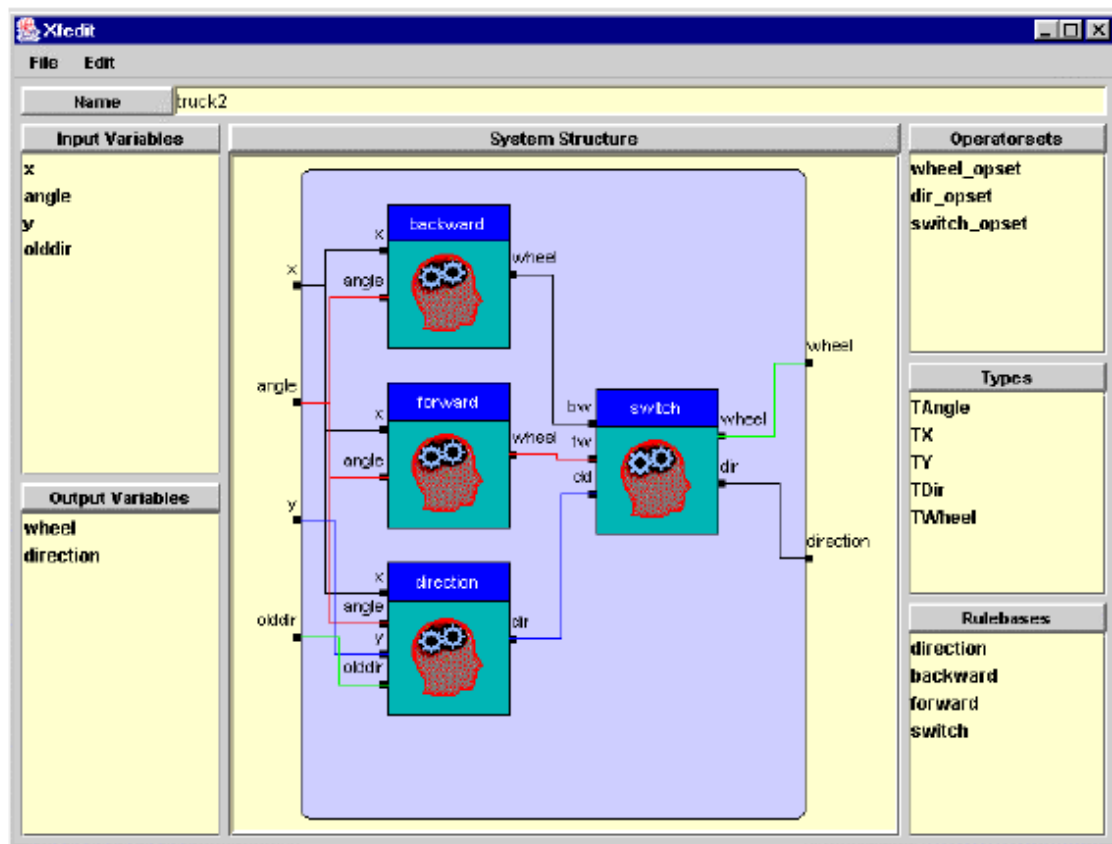
A continuación de la etiqueta using va el operador usado en la base de reglas.

Dentro del cuerpo de la definición de la base de reglas vemos que en cada línea vemos que pone factor ahí se introduce lo que pone en la primera columna de la vista libre de la pantalla de edición de la base de reglas.

Después vemos el antecedente de la regla y después el consecuente como se muestra en la vista libre de la base de reglas.

Entonces el sistema ya ha sido terminado, lo hemos editado con la herramienta Xfedit, que es la que nos resulto más sencillo su uso.

Se mostrara así nuestro sistema:

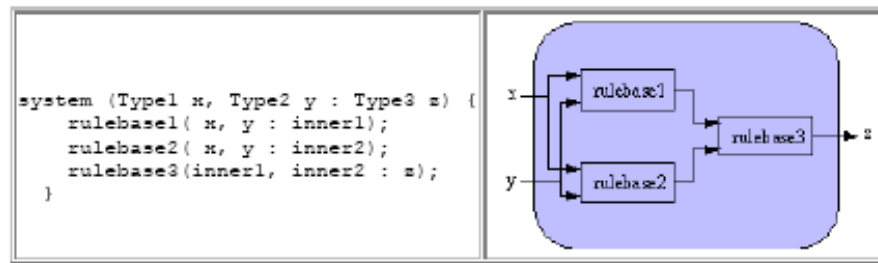


Para que el usuario entienda mejor el código del archivo de salida será como este formato:

```
system (input_list : output_list) {
  rule_base_identifier(inputs : outputs);
  rule_base_identifier(inputs : outputs);
  ..... }
```

Se pueden ver la lista de variables de entrada y la lista de variables de salida del sistema. Además se ven todas las reglas cada una con sus entradas y sus salidas.

Un ejemplo es el siguiente:



Se mostrara una pantalla como la siguiente en la que podemos comprobar todo lo que hemos introducido en nuestro sistema. Si queremos editar algo de lo introducido basta con señalarlo e irte al menú Edit y realizar el Edit oportuno. Si queremos eliminarlo solo debemos pulsar en lo que queremos eliminar e irnos al menú Edit y pulsar la opción Remove oportuno.

Luego nos sale una pantalla para confirmar, pulsamos Ok e inmediatamente realiza el sistema lo que le hayamos solicitado.

Para comprobar que el sistema realiza lo que el usuario desea, la herramienta XFuzzy proporciona una verificación del sistema.

En el menú de opciones Verification de la pantalla inicial de la herramienta se dan las siguientes opciones que iremos explicando una por una.

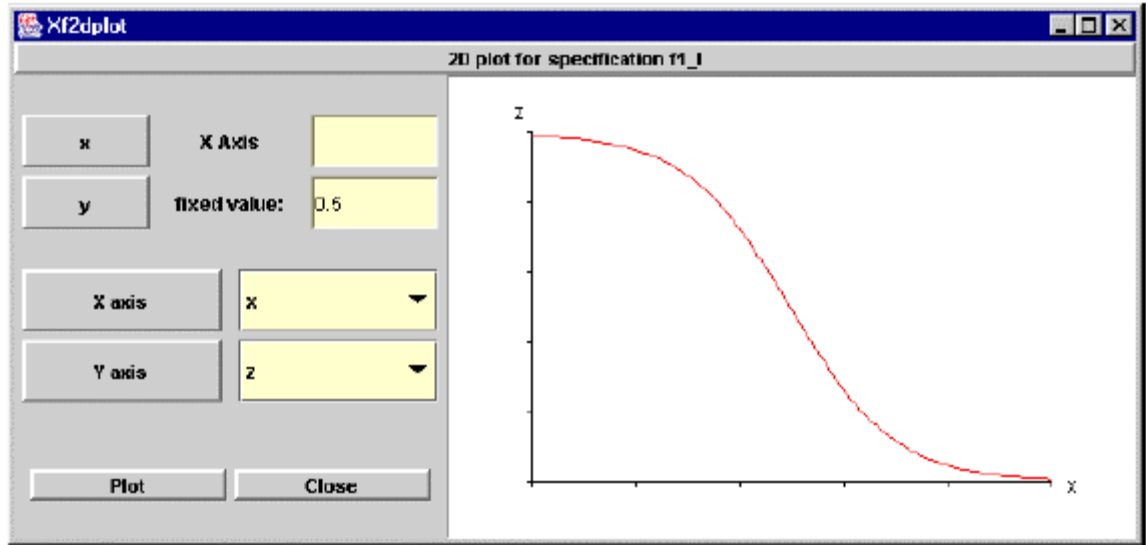
La opción xf2dplot que verifica el comportamiento de una variable de salida del sistema difuso en función de una variable de entrada.

En la ventana se muestran todas las variables de entrada al sistema y cuando tu eliges una lo debes de especificar donde pone X Axis. Después el valor de esta lo debes de poner en fixed value.

A continuación debes de poner la variable de salida del sistema y la especificas en Y Axis.

En la parte derecha de la pantalla se mostrara la grafica.



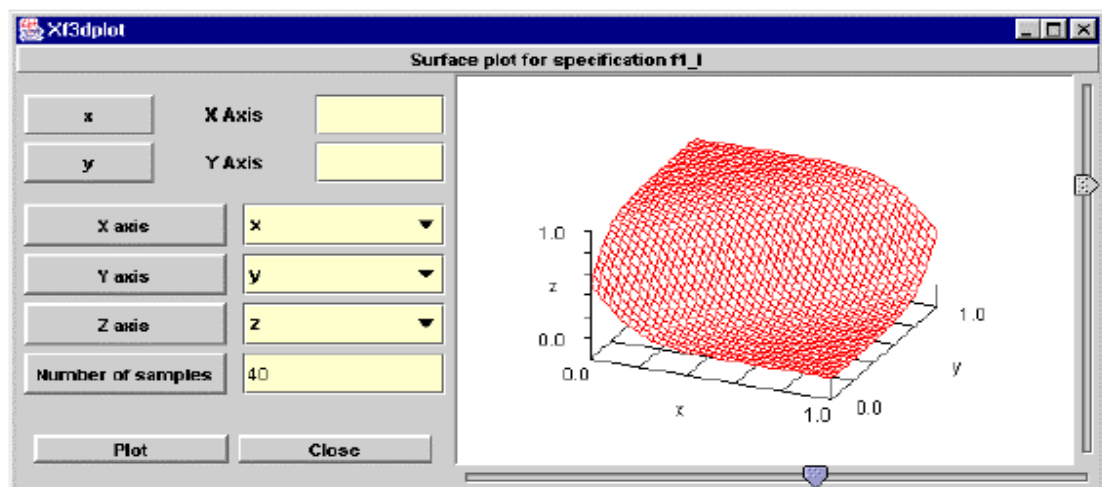


La opción xf3dplot que verifica el comportamiento de una variable de salida de un sistema difuso en función de dos variables de entrada.

Muestra una grafica tridimensional al contrario que la anterior que era bidimensional.

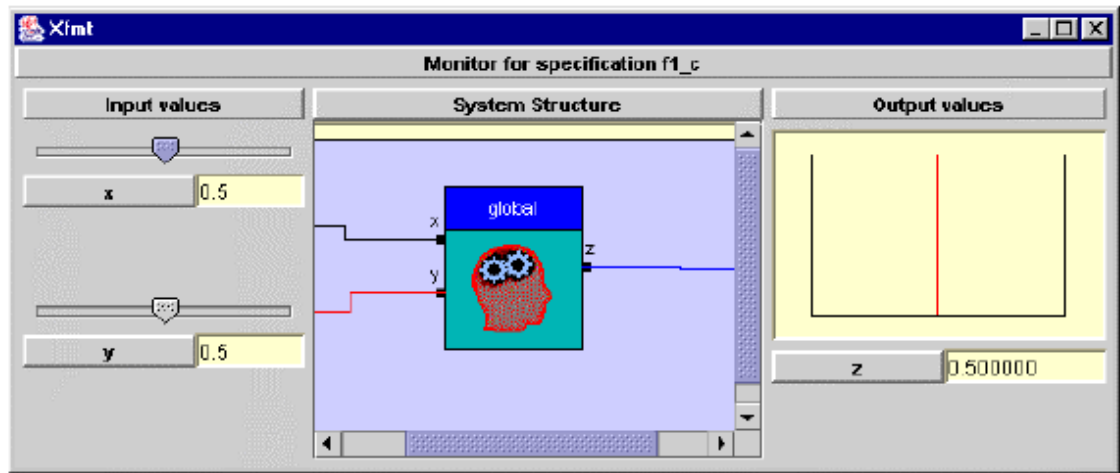
Introducimos la variable de entrada que sea la x en X Axis, la otra variable de entrada que sea la y en Y Axis con sus respectivos valores en fixed value correspondiente. Además debemos de introducir la Z en Z Axis que será la variable de salida del sistema.

Entonces se mostrara la grafica en la parte derecha de la pantalla.

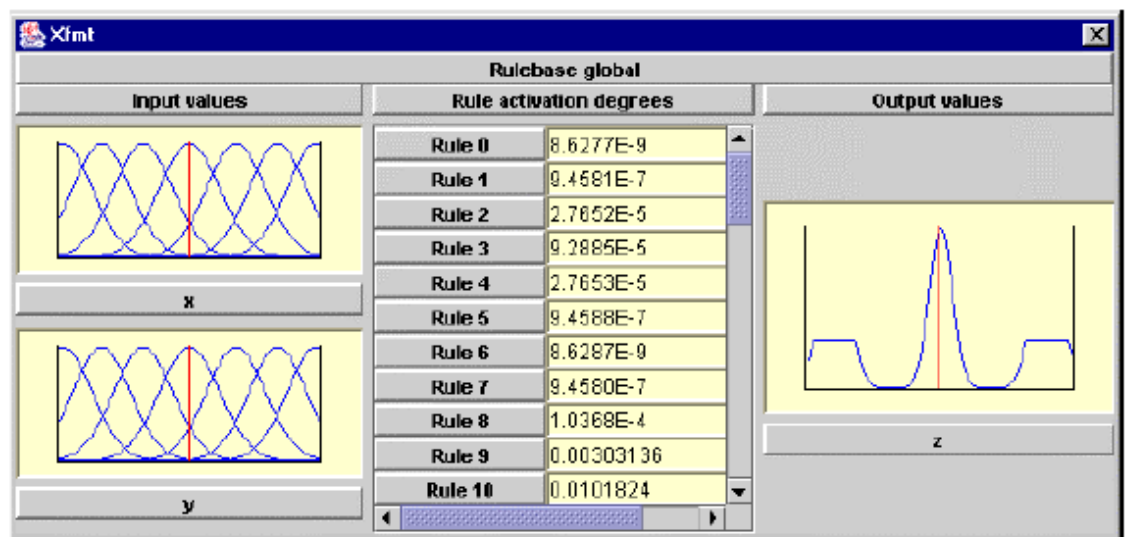


La opción de monitorización xfmt que lo que permite es elegir el valor de todas las variables de entrada al sistema y proporcionar los valores resultantes de la inferencia

borrosa sobre las variables de salida. En la parte izquierda de la pantalla se ven las distintas variables de entrada al sistema con el valor el cual podemos modificar. En la parte central el esquema de nuestro sistema global y en la parte derecha de la pantalla el valor de las variables de salida que tenga nuestro sistema así como la grafica correspondiente a cada una de estas.

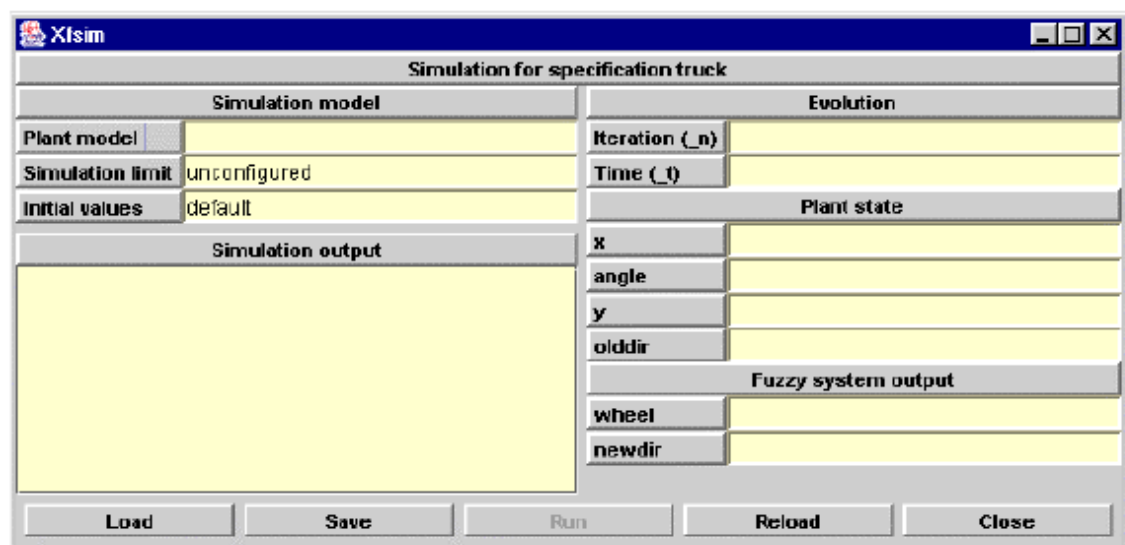


Para acceder a la ventana para monitorizar los valores internos del proceso de inferencia de cada base de reglas solo es necesario pulsar sobre la base de reglas en la representación del esquema perteneciente a cada base de reglas de nuestro sistema.



La opción de simulación xfsim que esta dirigida a estudiar los sistemas realimentados.

Esta opción no la explicamos ya que sería muy extensa su explicación, además de que su explicación se puede ver en el manual que se proporciona en la página web de la herramienta.



La herramienta XFuzzy también proporciona una etapa de síntesis que nos permite generar una implementación del sistema que pueda ser utilizada externamente.

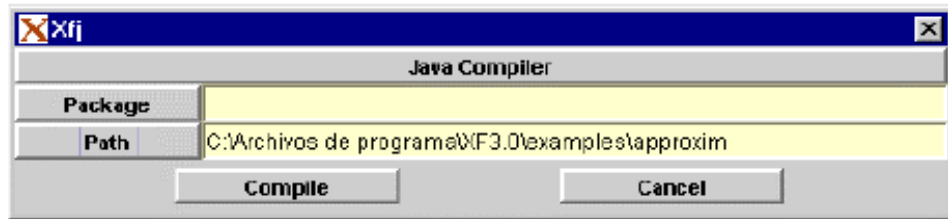
Existe la opción de crear esta implementación en lenguaje ANSI-C, en C++ y en lenguaje Java.

Para el lenguaje ANSI-C debemos de pulsar en el menú Síntesis y dar a la opción To C. Para el lenguaje C++ damos a la opción To C++ y finalmente para el lenguaje Java pulsar la opción Java.

En el manual explica más detenidamente las clases que proporciona cada una de estas opciones.

Nosotras usamos la herramienta de generación de código Java ya que nos resultara mucho más sencillo, ya que Java es uno de los lenguajes mas conocidos por nosotras.

Entonces pulsamos la opción To Java y nos sale una ventana como esta:



Los ficheros con código Java se generan en el mismo directorio que contiene al fichero del sistema por defecto.

Se generan los siguientes ficheros explicados en el manual de la página web y extraídos de allí textualmente.

*FuzzyInferenceEngine.java*, *MembershipFunction.java*, *FuzzySingleton.java* y *systemname.java*. Los tres primeros ficheros corresponden a descripciones de dos interfaces y una clase que son comunes a todos los sistemas de inferencia difusos. El último fichero contiene la descripción específica del sistema difuso *systemname.xfl*.

El fichero *FuzzyInferenceEngine.java* describe una interfaz Java que define un sistema de inferencia difuso general. Esta interfaz define cuatro métodos para implementar el proceso de inferencia con valores crisp y difusos.

```
public interface FuzzyInferenceEngine {
    public double[] crispInference(double[] input);
    public double[] crispInference(MembershipFunction[] input);
    public MembershipFunction[] fuzzyInference(double[] input);
    public MembershipFunction[] fuzzyInference(MembershipFunction[]
input);
}
```

El fichero *MembershipFunction.java* contiene la descripción de una interfaz usada para describir un número difuso. Contiene sólo un método, llamado *compute*, que calcula el grado de pertenencia para cada valor del universo de discurso del número difuso.

```
public interface MembershipFunction {
    public double compute(double x);
}
```

La clase *FuzzySingleton* implementa la interfaz *MembershipFunction*, que representa un valor crisp como un número difuso.

```
public class FuzzySingleton implements MembershipFunction {
    private double value;

    public FuzzySingleton(double value) { this.value = value; }
    public double getValue() { return this.value; }
    public double compute(double x) { return (x==value? 1.0: 0.0); }
}
```

Finalmente, el fichero *systemname.java* contiene la clase que describe el sistema difuso. Esta clase es una implementación de la interfaz *FuzzyInferenceEngine*. Por tanto, los métodos públicos que implementan la inferencia son los de la interfaz (*crispInference* y *fuzzyInference*).

## APÉNDICE: PROBLEMA CON LA HERRAMIENTA XFUZZY

Para que nos apareciesen las variables de entrada y las de salida cuando estas creando las rulebase tuvimos que modificar el fichero xfl que nos iba creando la herramienta Xfuzzy.

Tuvimos que hacer lo siguiente:

Cuando nos aparece esto en el documento xfl que nos genera la herramienta cogemos todas las variables de entrada, es decir, hasta los dos puntos desde el inicio del paréntesis después de system.

Lo que esta subrayado.

```
system (tBool inCaduca, tBool inPerenne, tBool inOpuesta, tBool inAlternativa, tBool  
inEstipulas, tBool inEntera, tBool inDividida, tBool inSimple, tBool inHermafrodita, tBool  
inUnisexual, tBool inHipogina, tBool inPerigina, tBool inEpigina, tBool inActinomorfa,  
tBool inSolitaria, tBool inIrregular, tBool inEsferoidal, tBool inGranos, tBool inSupero,  
tBool inInfero, tBool inBilabiada, tBool inEncapsulado, tBool inGlanduliferos, tBool  
inEstrellados, tBool inArbol, tBool inArbusto, tBool inLenosa, tBool inLiana, tBool  
inHerbacea, tBool inTrepadora, tBool inAcuatica, tBool inRizoma, tBool inIsosporeo,  
tBool inHeterosporeo, tPetalos inNumpetalos, tEstambres inEstam : tPlanta outPlantaProd,  
tPlanta outPlantaMin, tPlanta outPlantaLukas) {
```

Después para cada rulebase que creamos debemos de poner esto dentro del paréntesis que apareciera después del nombre que le hayamos dado a cada rulebase. En este caso mostramos rFamiliasProd.

En el podemos observar que hemos puesto todas las entradas al sistema igual que las subrayadas anteriormente.

También ponemos dos puntos seguido de estas variables y en cada caso ponemos alguna de las variables de salida que tenemos en el sistema, la que corresponda en cada caso. Para el producto la variable de salida es tPlanta outPlantaProd.

Para cada una de las tres rulebase que hemos generado nosotras hemos tenido que hacer exactamente lo mismo, pero en cada caso con la variable de salida correspondiente.

```
rulebase rFamiliasLukas (tBool inCaduca, tBool inPerenne, tBool inOpuesta, tBool
inAlternata, tBool inEstipulas, tBool inEntera, tBool inDividida, tBool inSimple, tBool
inHermafrodita, tBool inUnisexual, tBool inHipogina, tBool inPerigina, tBool inEpigina,
tBool inActinomorfa, tBool inSolitaria, tBool inIrregular, tBool inEsferoidal, tBool
inGranoso, tBool inSupero, tBool inInfero, tBool inBilabiado, tBool inEncapsulado, tBool
inGlandulíferos, tBool inEstrellados, tBool inArbol, tBool inArbusto, tBool inLenosa,
tBool inLiana, tBool inHerbacea, tBool inTrepadora, tBool inAcuática, tBool inRizoma,
tBool inIsosporo, tBool inHeterosporo, tPetalos inNumpetalos, tEstambres inEstam :
tPlanta outPlantaLukas) using lukas {
```

Vemos que pone using lukas eso es porque usa el operador lukas generado anteriormente.

## APÉNDICE: COMPARAR REGLAS XFUZZY CON LAS MODIFICADAS

```

_rl =
_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op
.and(_t_inPerenne.si.isEqual(inPerenne),_t_inOpuesta.si.isEqual(inOpuesta)),_t_inEstipula
s.no.isEqual(inEstipulas)),_t_inSimple.si.isEqual(inSimple)),_t_inHermafrodita.si.isEqual(i
nHermafrodita)),_t_inSolitaria.si.isEqual(inSolitaria)),_t_inIrregular.si.isEqual(inIrregular))
,_t_inSupero.si.isEqual(inSupero)),_t_inEncapsulado.si.isEqual(inEncapsulado)),_t_inGlan
duliferos.si.isEqual(inGlanduliferos)),_t_inArbusto.si.isEqual(inArbusto)),_t_inEstam.poco
s.isEqual(inEstam));

outPlantaProd.set(_i_outPlantaProd,_rl, _t_outPlantaProd.Acanthaceae);
_i_outPlantaProd++; sal [j] = _rl; j++;

```

Este es el código que resulta de la introducción de las reglas en el XFuzzy para la primera planta. Nosotras modificamos todas las reglas de la siguiente manera. Se pueden observar todas las modificaciones realizadas. Estas modificaciones están explicadas en la memoria.

```

_rl =
_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op
.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op
.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op.and(_op
.op.and(_op.and(_op.and(_op.and(
_op.or(_t_inPerenne.si.isEqual(inPerenne),noinfoPerenne),
_op.or(_t_inCaduca.no.isEqual(inCaduca),noinfoCaduca)),
_op.or(_t_inOpuesta.si.isEqual(inOpuesta),noinfoOpuesta)),
_op.or(_t_inAlternata.no.isEqual(inAlternata),noinfoAlternata)),
_op.or(_t_inEstipulas.no.isEqual(inEstipulas),noinfoEstipulas)),
_op.or(_t_inEntera.no.isEqual(inEntera),noinfoEntera)),
_op.or(_t_inDividida.no.isEqual(inDividida),noinfoDividida)),

```



\_op.or(\_t\_inSimple.si.isEqual(inSimple),noinfoSimple)),  
 \_op.or(\_t\_inHermafrodita.si.isEqual(inHermafrodita),noinfoHermafrodita)),  
 \_op.or(\_t\_inUnisexual.no.isEqual(inUnisexual),noinfoUnisexual)),  
 \_op.or(\_t\_inHipogina.no.isEqual(inHipogina),noinfoHipogina)),  
 \_op.or(\_t\_inPerigina.no.isEqual(inPerigina),noinfoPerigina)),  
 \_op.or(\_t\_inEpigina.no.isEqual(inEpigina),noinfoEpigina)),  
 \_op.or(\_t\_inActinomorfa.no.isEqual(inActinomorfa),noinfoActinomorfa)),  
 \_op.or(\_t\_inSolitaria.si.isEqual(inSolitaria),noinfoSolitaria)),  
 \_op.or(\_t\_inIrregular.si.isEqual(inIrregular),noinfoIrregular)),  
 \_op.or(\_t\_inEsferoidal.no.isEqual(inEsferoidal),noinfoEsferoidal)),  
 \_op.or(\_t\_inGranos.no.isEqual(inGranos),noinfoGranos)),  
 \_op.or(\_t\_inSupero.si.isEqual(inSupero),noinfoSupero)),  
 \_op.or(\_t\_inInfero.no.isEqual(inInfero),noinfoInfero)),  
 \_op.or(\_t\_inBilabiada.no.isEqual(inBilabiada),noinfoBilabiada)),  
 \_op.or(\_t\_inEncapsulado.si.isEqual(inEncapsulado),noinfoEncapsulado)),  
 \_op.or(\_op.or(\_t\_inGlanduliferos.si.isEqual(inGlanduliferos),\_t\_inGlanduliferos.no.  
 isEqual(inGlanduliferos)),noinfoGlanduliferos)),  
 \_op.or(\_t\_inEstrellados.no.isEqual(inEstrellados),noinfoEstrellados)),  
 \_op.or(\_t\_inArbol.no.isEqual(inArbol),noinfoArbol)),  
 \_op.or(\_t\_inArbusto.si.isEqual(inArbusto),noinfoArbusto)),  
 \_op.or(\_t\_inLenosa.no.isEqual(inLenosa),noinfoLenosa)),  
 \_op.or(\_t\_inLiana.no.isEqual(inLiana),noinfoLiana)),  
 \_op.or(\_t\_inHerbacea.no.isEqual(inHerbacea),noinfoHerbacea)),  
 \_op.or(\_t\_inTrepadora.no.isEqual(inTrepadora),noinfoTrepadora)),  
 \_op.or(\_t\_inAcuatica.no.isEqual(inAcuatica),noinfoAcuatica)),  
 \_op.or(\_t\_inRizoma.no.isEqual(inRizoma),noinfoRizoma)),  
 \_op.or(\_t\_inIsosporeo.no.isEqual(inIsosporeo),noinfoIsosporeo)),  
 \_op.or(\_t\_inHeterosporeo.no.isEqual(inHeterosporeo),noinfoHeterosporeo)),  
 \_op.or(0,noinfoNumpetalos)),  
 \_op.or(\_t\_inEstam.pocos.isEqual(inEstam),noinfoEstam));

```
outPlantaProd.set(_i_outPlantaProd,_rl, _t_outPlantaProd.Acanthaceae);  
_i_outPlantaProd++;  
sal [j] = _rl; j++;
```

## **PALABRAS CLAVE**

**New 165, 167, 169, 170, 179, 180, 186, 249, 251, 257, 261, 262, 263**

**Experto 5, 22, 33, 46, 142, 152, 161, 225, 227**

**Biólogo 5, 34, 41, 42, 47, 93, 224, 231, 232, 239**

**Rulebase 180, 181, 183, 184, 190, 197, 198, 263, 267, 276, 277, 281, 288, 296**

**Ejecución 122, 128, 131, 194, 198, 199, 208, 209, 210, 217, 223, 225, 231, 235, 236, 239**

**Pioneras 45, 46, 138**

**Nítida 34, 45, 111, 199, 200, 201, 202, 204, 212, 214, 217, 219, 222, 237**

**Conocimiento 7, 17, 33, 44, 46, 138, 140, 141, 152, 161, 185, 225, 227, 247**

**Dicotómico 42, 43, 44, 138**

# BIBLIOGRAFÍA

- SALVAT UNIVERSAL DICCIONARIO ENCICLOPÉDICO

Para las descripciones de los términos específicos del campo tratado, las familias.

- <http://www.rjb.csic.es/floraiberica/>  
La primera de las determinaciones mostrada.
- <http://www.floradeiberia.com/determinación.php>  
La segunda determinación explicada.
- <http://www.botanical-online.com/lasflores.htm>  
Para las definiciones de los características de las familias.
- <http://www.botanical-online.com/spanishglossary.htm>  
Para las definiciones de las características de las familias.
- [http://www.imse.cnm.es/Xfuzzy/index\\_sp.html](http://www.imse.cnm.es/Xfuzzy/index_sp.html)  
Para el manual y la descarga de la herramienta de lógica borrosa usada.
- <http://jardinactual.com/articuloshtm2.php?articulo=308>
- <http://www.florachilena.cl/Glosario/glosario.htm>  
Para los términos añadidos por el biólogo.
- <http://www.rjb.csic.es/floraiberica/floraiberica/texto/pdfs/000%20clavegeneral.pdf>  
Documento proporcionado por el experto biólogo para mejorar la determinación de las plantas.
- <http://www.geocities.com/icatercera/electronica/fuzzy.html>  
Para la explicación de la lógica borrosa.